8-15-2014

Security and Privacy in Online Social Networks

Arun Thapa

Security and privacy in online social networks

By

Arun Thapa

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Electrical Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

August 2014

Security and privacy in online social networks

By

Arun Thapa

Approved:

Pan Li
(Major Professor)

Qian (Jenny) Du
(Committee Member)

James E. Fowler
(Committee Member and Graduate Coordinator)

Erdem Topsakal
(Committee Member)

Jason Keith
Interim Dean
Bagley College of Engineering

Name: Arun Thapa

Date of Degree: August 15, 2014

Institution: Mississippi State University

Major Field: Electrical Engineering

Major Professor: Dr. Pan Li

Title of Study: Security and privacy in online social networks

Pages of Study: 122

Candidate for Degree of Doctor of Philosophy

The explosive growth of Online Social Networks (OSNs) over the past few years has redefined the way people interact with existing friends and especially make new friends. OSNs have also become a great new marketplace for trade among the users. However, the associated privacy risks make users vulnerable to severe privacy threats. In this dissertation, we design protocols for private distributed social proximity matching and a private distributed auction based marketplace framework for OSNs.

In particular, an OSN user looks for matching profile attributes when trying to broaden his/her social circle. However, revealing private attributes is a potential privacy threat. Distributed private profile matching in OSNs mainly involves using cryptographic tools to compute profile attributes matching privately such that no participating user knows more than the common profile attributes. In this work, we define a new asymmetric distributed social proximity measure between two users in an OSN by taking into account the weighted profile attributes (communities) of the users and that of their friends'. For users with

different privacy requirements, we design three private proximity matching protocols with increasing privacy levels. Our protocol with highest privacy level ensures that each user's proximity threshold is satisfied before revealing any matching information.

The use of e-commerce has exploded in the last decade along with the associated security and privacy risks. Frequent security breaches in the e-commerce service providers' centralized servers compromise consumers' sensitive private and financial information. Besides, a consumer's purchase history stored in those servers can be used to reconstruct the consumer's profile and for a variety of other privacy intrusive purposes like directed marketing. To this end, we propose a secure and private distributed auction framework called SPA, based on decentralized online social networks (DOSNs) for the first time in the literature. The participants in SPA require no trust among each other, trade anonymously, and the security and privacy of the auction is guaranteed. The efficiency, in terms of communication and computation, of proposed private auction protocol is at least an order of magnitude better than existing distributed private auction protocols and is suitable for marketplace with large number of participants.

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. Pan Li for his guidance during my graduate study at Mississippi State University. His hard work, patience, motivation, enthusiasm, and immense knowledge has inspired me to become not only a better researcher but also a better person in life. I would like to thank my fellow colleagues in the Network, Energy, Security, and big daTa (NEST) research group. I have always cherished the academic/non-academic discussions and jokes in the group.

I would like to acknowledge the support of my committee members Dr. Qian (Jenny) Du, Dr. James E. Fowler, and Dr. Erdem Topsakal. Thank you for serving as my committee members. I would also like to thank the Department of Electrical and Computer Engineering, Mississippi State University, for supporting me through my graduate studies and in my hard time.

I am grateful to my friends in Starkville and everyone who helped me get through my hard time.

Last but not the least, I would like to thank my beautiful wife Sabina and my loving family for believing in me and supporting me all the time. I would not be where I am today without their continuous support and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The Internet incorporated security and privacy as an afterthought in its design. The awareness and sophistication in technology for online security and privacy have seen tremendous improvement over the past decade. However, the threats for online security and privacy have also evolved and become more challenging than ever before. The sophistication at which attackers compromise users' sensitive information has improved significantly over the years. The centralized servers of the Internet service providers represent single point of failure and have been attacked frequently in the past compromising millions of customers' private data. Besides, they have also been used as tools for mass surveillance by the government organizations. In addition to malicious hackers/attackers stealing users' private information, primarily with financial motive, the changing threat model landscape also includes Advanced Persistent Threat (APT) and global surveillance capability from very resourceful entities and organizations. The future system design need to be secure against such challenging and evolving threat models. The need for better online security and privacy against such threat models has brought several decentralized solutions from academia. Tor [21] is one such solutions which has been used widely as anonymous browsing tool to tackle Internet surveillance in hostile environment and to protect privacy of users. Peer to

peer crypto currency Bitcoin [57] is a decentralized electronic cash that may have potential to disrupt traditional centralized banking system. Similarly, Diaspora [1] is a decentralized online social network (DOSN), where users own their data and store on their own servers or on the servers they trust. Motivated by such robust decentralized solutions, in this dissertation, we propose protocols for private social proximity matching based on a novel asymmetric social proximity measure, and a secure and private distributed auction framework called SPA, for the first time in the literature, based on online social networks.

The explosive growth of Online Social Networks (OSNs) over the past decade has dramatically changed the way people communicate with their peers, consume/produce information, conduct business, and expand their social circles. The ease at which people communicate with their peers and to the world has challenged the traditional unidirectional information flow paradigm, where information (e.g., breaking news) flows mainly through a source (e.g., a news organization) to the consumers. The new multidirectional information flow paradigm, where users of OSNs (e.g., Facebook, Twitter, YouTube) are both the sources and and the consumers of the information, has benefited ordinary users in many ways. People find it easier to share their ideas in their friend circles and to connect to new people, who share similar interests. Besides, the rich connectivity of OSNs provides a new avenue for e-commerce among the users. However, the centralized architecture of the traditional OSNs poses serious privacy threats to users. The central servers store and monetize on users' sensitive private information and become a single point of failure, which have been breached by malicious attackers [12, 37] frequently leading to compromised users' private data. The motivation of preserving users' privacy has led to

privacy preserving decentralized OSN architectures [1, 8, 18], where users store their private data on their own servers or on the servers they trust. In this dissertation work, we design protocols for private social proximity matching between two users in an OSN based on a novel asymmetric social proximity measure.

People look for shared attributes (e.g., interests, geography, friends, and religion) when initiating/responding to new friendship requests. A naive way of matching the attributes is to list each one's attributes and find the match. However, people are reluctant to reveal their attributes to a stranger. A Trusted Third Party (TTP) can help match the attributes if it is trusted to hold everybody's attributes securely and return the matching attributes upon request by the users. Similar to the central servers in the traditional OSNs, the TTP stores the attributes of each user and represents a single point of failure. Besides, people are increasingly hesitant to store their personal information in a centralized server or TTP. The problem of matching attributes between two users privately without a TTP can be solved with the help of cryptography. Users with private attributes set can take part in cryptographic protocol transactions to find the common attributes among them without revealing additional information about their attributes to one another. There have been several works in cryptography, which can be used to find the intersection of the input set (or some function thereof) of participating users. In this work we define an asymmetric social proximity metric between two users based on weighted attributes of the users and their friends'. We then propose three protocols for privately matching the social proximity between two users for users with varying degree of privacy requirements. In our protocol with the highest privacy level, participating users make sure the social proximity metric,

defined independently by each of them, satisfies their own private thresholds before letting the other user know the matching information. The protocols are introduced in details with our specific contributions in Chapter 2.

E-commerce has become an important part of commerce in today's world. It enables easy access to goods and services via the Internet. However, the security and privacy risks associated with e-commerce are also on the rise. Attackers are frequently able to compromise the e-commerce service providers' secure servers and steal consumers' private information. A consumer's history of sales/purchases stored in those servers is critically sensitive private data, which can be exploited for many privacy intrusive purposes like directed marketing and more importantly consumer profiling. Moreover, in auction based e-commerce like eBay, consumers need to trust on the honesty of the auctioneer (server). The bidding statistics also reveals important private information about the users' valuation of the goods. If malicious, the server may increase its financial gain by exploiting the bidding statistics and colluding with sellers. In light of these security and privacy threats, we envision a distributed private marketplace framework, where users do not need to trust any third party servers and can trade anonymously with bid privacy and auction correctness guarantee. In this regard, we propose SPA: a Secure and Private Auction framework based on decentralized online social networks, which, to the best of our knowledge, is the first protocol of its kind for online social networks in the literature. We present the proposed framework in detail in Chapter 3.

The rest of the proposal is organized as follows. In the next Chapter (Chapter 2), we detail our proposed asymmetric social proximity measure and cryptographic protocols for

private social proximity matching. The proposed SPA framework is presented in Chpater 3 with real social network data set experiment and simulations to verify the cost of computation and communication. Finally, in Chapter 4 we conclude our dissertation with future plans.

CHAPTER 2

ASYMMETRIC SOCIAL PROXIMITY BASED PRIVATE MATCHING PROTOCOLS FOR ONLINE SOCIAL NETWORKS

2.1 Introduction

Online Social Networks (OSNs) have had tremendous growth over the past few years. OSNs such as Facebook, Google$^+$, Linkedin are some of the most visited sites on the Internet [4], where users spend a significant fraction of their online time. Besides, increasing popularity of smart phones has extended the platforms used for accessing online social networks and provided a plethora of opportunities for mobile social networking. OSNs have redefined the way people interact with existing friends, and more importantly, make new friends. In particular, people can now explore potential friendships via OSNs, by looking for common interests, friends, and symptoms, close geographic proximity, etc., between each other. A naive solution to finding new friends in OSNs is using a server that stores all the users' information and conducting profile matching through the server. In this case, however, the server knows all the users' private information and becomes a single point of failure. Thus, if the server gets compromised, all users' privacy is at risk. For example, Twitter was attacked in early January 2013 and about 250,000 user accounts might have been compromised, with names and e-mails possibly being uncovered [12]. Facebook, Apple, Microsoft were under similar attacks in February 2013 [37]. Moreover, users may

not have connectivity to the server all the time. Therefore, there has been growing interests in new privacy-preserving distributed solutions to finding friends in OSNs.

In OSNs and Mobile Social Networks (MSNs), many distributed solutions to privately finding the social proximity between two users have been proposed. The most common way of determining friendship between two people is through profile matching, i.e. finding out if they have common profile attributes, like interests [50], [75], symptoms [46, 47, 53], or some other social coordinates [22, 77]. In some cases, the number of common friends also serves as the proximity measure between two users [16], [56]. Such previous works employ various cryptographic tools to protect the privacy of the profile information of the users in the private matching process. After two strangers, say with profile attribute sets X and Y, execute a private matching protocol, the one who initiates the protocol will know either $X \cap Y$ or some function of $X \cap Y$ while the other one who responds does not know anything. Thus, a malicious user can execute the protocol with any user and leave without letting him/her do the same.

Moreover, most previous schemes for profile matching in online/mobile social networking are based on the premise that two people are likely to establish a social relationship only if they share similar profile attributes like interests, symptoms, or some other social coordinates. While it is true that people with similar profile attributes are likely to be friends, this is not the only way of determining friendship. For example, a doctor's best friend may not necessarily always be a doctor, but can be a writer who may share very few common profile attributes. In another example, two students who both have a lot of good friends studying in the Electrical and Computer Engineering department may become

good friends, although they do not share many common profile attributes. We notice that whether two people can become friends not only depends on whether they have anything in common, but also is affected by whether their friends have anything in common. The intuition behind this is simple: a friend's friend can also be a friend.

In this work, we leverage community structures to redefine the OSN model, and propose an asymmetric social proximity between two users. In particular, we consider that each OSN user is affiliated with some communities (or groups)[1], which the user weighs differently. We notice that the communities can actually tell a lot about their members. There can be a wide variety of communities in an OSN like a university community, a department community, a fan community of an artist, movies, or sports, and a community of certain professions. Besides, we notice that in real life people also value their friendships differently. Thus, we propose an asymmetric social proximity between two users, which is the cumulative weight of the common communities to one user considering both his/her and his/her friends' perceptions. We also design three different private matching protocols based on the proposed asymmetric social proximity. The main contributions of this work can be briefly summarized as follows.

- We define an *asymmetric social proximity* measure between two users in an OSN, which considers both each user's and his/her friends' perceptions on the common communities between the two users. This proposed asymmetric social proximity can better capture the characteristics of making friends in OSNs.
- Based on the asymmetric social proximity, we design three different private matching protocols, i.e., L1P, L2P/EL2P, and L3P, which provide users with different privacy levels. In particular, our protocol L3P with the highest privacy level ensures that two users will not know any of their common communities before they become friends.

[1]In what follows, we use "communities" and "groups" interchangeably.

- We analyze the privacy, and computation and communication cost of the proposed protocols. Our protocols protect users' privacy better than the previous works based on symptoms, interests, and the number of common friends, with lower computation and communication cost. Particularly, in most previous schemes, e.g., [16, 22, 46, 47, 50, 53, 75], a malicious user A can request friendship with another user B and then leave with B's private information before B knows anything about A. In our schemes, when one maliciousx user A requests friendship with another user B, A can know some limited private information of B's only if B is willing to accept the request.
- We validate our proposed asymmetric proximity measure using real social network data and conduct extensive simulations to evaluate the performance of the proposed protocols in terms of computation cost, communication cost, total running time, and energy consumption. The results show the advantages of our protocols over state-of-the-art protocols.

The rest of the Chapter is organized as follow. We discuss the related works in Section 2.2 and present our system model in Section 2.3. We detail the proposed three asymmetric social proximity based private matching protocols in Section 2.4. We present simulation results in Section 2.5, and finally conclude the Chapter in Section 2.6.

2.2 Related Work

In this section, we briefly introduce some previous studies that are most relevant to our work.

2.2.1 Private Set Intersection (PSI) protocols

In PSI protocols, two or more parties carrying their respective input sets interact to privately find the intersection set. In a two party (a server and a client) PSI protocol, the two parties interact so that the client learns only the intersection of the two input sets and the size of the server's input set, while the server learns nothing but the size of the client's input set. Since the introductory work of Freedman, Nissim, and Pinkas (FNP) [28], several

PSI protocols [17, 27, 34, 35, 38, 43] secure under semi-honest and/or malicious adversary models have been proposed. In such schemes, a client can artificially inflate its input set to learn the server's whole input set. Authorized PSI (APSI) protocols [9, 10, 15] avoid this problem by verifying the participants' inputs using some trusted authority. They involve expensive cryptographic processes, which lay heavy burdens on users' mobile devices.

2.2.2 Secure Multiparty Computation (SMC) protocols

SMC protocols allow two or more parties to privately calculate some functions of their inputs such that no party knows more than the function output and its own input. In particular, Yao [72] proposes the first SMC protocol based on garbled circuits. After that, there are a lot of works on improving security [31] and/or computation and communication complexities [5, 19, 20, 26, 30, 36]. We do not employ SMC schemes for private proximity measurement in our scheme for two reasons. First, the generic SMC protocols are prohibitively expensive in both communication and computation. Second, our proposed social proximity measurement involves not only the users inputs (i.e., communities), but also their private parameters (i.e., α's and β's that will be introduced Section 2.3) which cannot be fed into the circuits to calculate proximity.

2.2.3 Social Proximity

The graph structure of social networks has been exploited to derive effective proximity measures. Katz measure [41] uses an ensemble of all the paths between two users in the network graph to derive the social proximity. Liben-Nowell et al. [48] and Tong et al. [67] also employ path-ensemble based methods for the future link prediction in social

networks and proximity measurement. The path-ensemble based proximity measures are known to be effective in link prediction and proximity measurement in social networks as they capture more information about the underlying social network. However, they require the knowledge of the snapshot of the social network graph, and are prohibitively expensive in computation. Thus, these methods are not applicable to distributed proximity measurement.

2.2.4 Social Proximity Based Private Matching

Among distributed measurements of social proximity, one of the most common and simplest proximity measure is the number of common friends or profile attributes between two users of the network [16, 22, 46, 47, 50, 53]. Intuitively, as the overlap between two user's profile attributes or friend spaces grows, their proximity increases. Based on distributed social proximity measurement, Zhang et al. [75] use homomorphic encryption to obtain fine-grained profile matching for mobile social networks. Similar profile matching schemes are presented in [16, 22, 46, 50, 77]. Profile matching in mobile health social network is studied in [47, 53] to privately match health profiles. Recently, Zhang et al. [74] proposes a mechanism to match-making profile search in a decentralized multi-hop mobile social network, where a user submits his/her "preference-profile" in order to search other users matching the profile. Similarly, Nagy et al. [56] presents a framework for finding common friends in a private manner using secure computation, set intersection, and bloom filters. Note that most of these studies focus on profile matching under the assumption that the social proximity between two users is symmetric, i.e., the social proximity calculated

by each user is the same. In this work, we utilize the communities and friend circles in an OSN to derive a realistic asymmetric social proximity in a distributed manner.

2.3 System Model

2.3.1 Network Model

Consider an online social network (OSN) where users store their own and friends' information on their devices. Such an OSN can be a decentralized OSN like that in [1], where no single server has information about all users, and two users can communicate via the Internet to establish a friendship. It can also be an MSN where two users' in close proximity can utilize bluetooth or WiFi to communicate for private matching. In addition, the network considered herein also includes the scenarios in centralized OSNs like Facebook, Google+, where users may not always be connected to the servers and can use the information stored in their mobile devices to find friends without the servers' involvement.

Note that we consider social friendships bidirectional, mutual, and reciprocating. In other words, if A is a friend of B's, B is also a friend of A's. Besides, we notice that in real life people value their friendships differently. Thus, as shown in Figure 2.1, we propose that each user groups his/her friends into different friend circles like hometown friends, family friends, university friends, co-workers, and gym friends. In addition, we consider that each user i is affiliated with a set of communities, denoted by $C_i = \{C_i^1, C_i^2, ..., C_i^{c_i}\}$. The whole set of communities a user i or his/her friends are affiliated with, called "the overall community set" of user i and denoted by $\overline{C}_i$, is $\overline{C}_i = \bigcup_{j \in \overline{\mathcal{N}}_i} C_j$, where $\overline{\mathcal{N}}_i = \mathcal{N}_i \cup \{i\}$ and

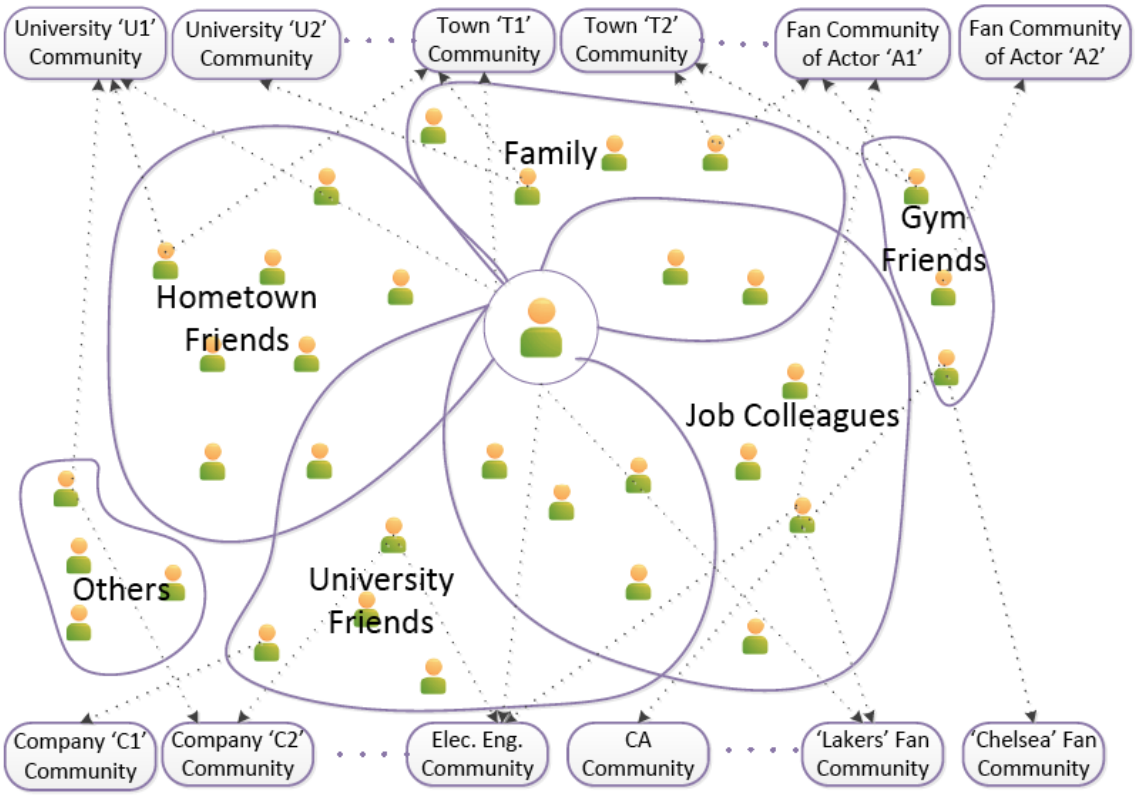

Figure 2.1

System Model

$\mathcal{N}_i$ denotes the set of user i's friends. We call a community in $\overline{C}_i$ one of user i's overall communities.

2.3.2 Asymmetric Distributed Social Proximity Measurement

In order to measure the social proximity (denoted by Ψ) between two users in an OSN without revealing their privacy, we utilize the users' overall community sets instead of their private profiles. The intuition behind this is that two persons who both have a lot of close friends in the same several communities can probably be friends. In particular, we take the following parameters into account. First, as mentioned before, a user in an OSN divides his/her friends into different friend circles, which represent different friendship weights to the user. In particular, suppose a user i has a set of friend circles $FC_i = \{FC_i^1, FC_i^2, ..., FC_i^{f_i}\}$. In order to quantify the significance of a particular friend circle FC_i^j ($1 \leq j \leq f_i$), user i assigns an integer value α_i^j ($0 \leq \alpha_i^j \leq \alpha^{max}$) to FC_i^j. A larger α_i^j indicates higher importance of the friend circle to the user. Second, each user, say i, also assigns an integer weight factor to each of the communities he/she is affiliated with, say C_i^j ($1 \leq j \leq c_i$), which is denoted by $\beta_i(C_i^j)$ ($0 \leq \beta_i^j \leq \beta^{max}$). Note that α^{max} and β^{max} are predefined system parameters (integers) that are known to all the users.

Considering the above parameters, we define a community based social proximity between two users A and B as follows. Let $C_{AB} = \overline{C}_A \cap \overline{C}_B = \{C_{AB}^1, C_{AB}^2, ..., C_{AB}^{c_{AB}}\}$, and $FC(i, j)$ denote a function which returns user i's friend circle(s) that i's friend, j, is in,

i.e., $j \in FC_i^k$ for any $k \in FC(i,j)$. Besides, we define $FC(i,i) = \{0\}$ for any i, and $\alpha_j^0 = \alpha^{max}$ for any j. Thus, the social proximity between A and B gauged by A is

$$\Psi_{A\leftarrow B} = \frac{\sum_{i=1}^{|C_{AB}|} \sum_{j\in C_{AB}^i \cap \overline{\mathcal{N}}_A} \left(\beta_j(C_{AB}^i) \sum_{\{k|k\in FC(A,j)\}} \left(\alpha_A^k \right) \right)}{\sum_{i=1}^{|\overline{C}_A|} \sum_{j\in C_A^i \cap \overline{\mathcal{N}}_A} \left(\beta_j(C_A^i) \sum_{\{k|k\in FC(A,j)\}} \left(\alpha_A^k \right) \right) \left(\right.}, \tag{2.1}$$

and that gauged by B is

$$\Psi_{B\leftarrow A} = \frac{\sum_{i=1}^{|C_{AB}|} \sum_{j\in C_{AB}^i \cap \overline{\mathcal{N}}_B} \left(\beta_j(C_{AB}^i) \sum_{\{k|k\in FC(B,j)\}} \left(\alpha_B^k \right) \right)}{\sum_{i=1}^{|\overline{C}_B|} \sum_{j\in C_B^i \cap \overline{\mathcal{N}}_B} \left(\beta_j(C_B^i) \sum_{\{k|k\in FC(B,j)\}} \left(\alpha_B^k \right) \right) \left(\right.}, \tag{2.2}$$

where $0 \leq \Psi_{A\leftarrow B} \leq 1$ and $0 \leq \Psi_{B\leftarrow A} \leq 1$. Evidently, the social proximity measures defined above are rational numbers. In (2.1), $\sum_{\{k|k\in FC(A,j)\}} \alpha_A^k$ is the total weight of friend j to A considering the multiple friend circles of A that j is in. $\beta_j(C_{AB}^i)$ is the weight of one of the common communities, i.e., C_{AB}^i, shared by A and B to j. Thus, the numerator of the right-hand-side (RHS) of (2.1) represents the total equivalent weight of the common communities shared by A and B to A, considering both A's and A's friends' perceptions. Similarly, the denominator of the right-hand-side (RHS) of (2.1) represents the total equivalent weight of A's communities to A, considering both A's and A's friends' perceptions. Thus, (2.1) is the normalized weight of the common communities to A considering both A's and A's friends' perceptions. In other words, (2.1) quantifies how important the common communities are to A. Similarly, (2.2) is the normalized weight of the common communities to B considering both B's and B's friends' perceptions and quantifies how important the common communities are to B. Notice that when calculating $\Psi_{A\leftarrow B}$, A only needs his/her

weights on his/her own friend circles and his/her friends' weights on As communities. In general, a larger $\Psi_{A \leftarrow B}$ indicates a closer social relationship of B to A.

Note that the proposed social proximity measurement is asymmetric, i.e., $\Psi_{A \leftarrow B}$ and $\Psi_{B \leftarrow A}$ are not necessarily equal. This is different from most of the distributed proximity measurements proposed for private matching, which are symmetric. We contend that asymmetric social proximity is more realistic, which is supported by a common intuition that the fact that A is the best friend of B does not necessarily mean B is the best friend of A.

2.3.3 Cryptographic Tools

2.3.3.1 Paillier Cryptosystem

Paillier designed an efficient asymmetric cryptosystem, called Paillier cryptosystem [60], based on decisional composite residuosity assumption. Due to its attractive additive homomorphic property, Paillier cryptosystem has been widely used in many applications like secure e-voting and private information retrieval. In particular, letting $\mathrm{ENC}(\cdot)$ and $\mathrm{DEC}(\cdot)$ denote the encryption and decryption functions of Paillier scheme, respectively, we have

- $\mathrm{ENC}(m_1) \cdot \mathrm{ENC}(m_2) = \mathrm{ENC}(m_1 + m_2)$
- $\mathrm{ENC}(m)^c = \mathrm{ENC}(c \cdot m)$

The Paillier cryptosystem is semantically secure for sufficiently large public keys, which means that it is infeasible for a computationally bounded adversary to derive significant information about a message (plaintext) when given only its ciphertext and the corresponding

public key. In this study, we assume that the public key is of 1184 bits for sufficient semantical security of the Paillier cryptosystem [75]. Therefore, a ciphertext is of 2048 bits, a Paillier encryption needs two 1024-bit exponentiations and one 2048-bit multiplication, and a Paillier decryption's cost is equivalent to one 2048-bit exponentiation.

Note that the proposed protocols can in fact work with any cryptosystem which is semantically secure and supports additive homomorphism. We employ Paillier cryptosystem to facilitate our illustrations in this work.

2.3.3.2 The FNP Scheme

Freedman et al. [28] design a private matching and set intersection protocol, called FNP, using homomorphic encryption, on which our protocols are based. In FNP, a client constructs a polynomial $P(z) = (x_1 - z)(x_2 - z).....(x_{n_C} - z) = \sum_{k=0}^{n_C} u_k z^k$, where $x_1, x_2, ..., x_{n_C}$ are the elements of the client's input set X. The client then encrypts the coefficients and send $\mathrm{ENC}(u_0), \mathrm{ENC}(u_1), ..., \mathrm{ENC}(u_{n_c})$ to the server. Utilizing the homomorphic property, the server constructs and evaluates the encrypted polynomial $\mathrm{ENC}(P(z))$ at each of the element in its own input set Y. The server then chooses a random number ρ_i, and computes and returns to the client $\mathrm{ENC}(\rho_i P(y_i) + y_i)$ for each $y_i \in Y$. When the client decrypts the ciphertext received from the server, it can find all $y_i \in X \cap Y$ as $P(y) = 0$ for all y_i's which are the roots of the polynomial $P(z)$ constructed by the client.

2.3.4 Adversary Model

Although there could be outsider adversaries trying to eavesdrop on the communications in the OSN, or modify, replay and inject messages, we focus on insider adversaries

in our protocol design, who are the participators of the protocols and pose more challenges in protecting users' privacy. We believe, in the context of social networks, semi-honest or Honest But Curious (HBC) adversary model best describes the characteristics of adversaries, which is considered as the adversary model in this study. A semi-honest adversary faithfully executes the protocols correctly but at the same time tries to gather more information about the other party than the protocols intend to disseminate.

2.4 Asymmetric Social Proximity Based Private Matching Protocols

In this section, we propose three novel and efficient social proximity based private matching protocols with different privacy levels. Before we delve into details, we first present some definitions below.

- **Initiator**[2]: An Initiator is an OSN user who initiates a protocol for calculating social proximity. In other words, an Initiator is an OSN user who asks another user (a Responder) for friendship.
- **Responder**: A Responder, upon the the request from an Initiator, replies by following the protocol.

Besides, when an Initiator asks a Responder for friendship, it should be the Responder who determines whether or not to accept the request by executing the protocol to find the social proximity.

2.4.1 Protocol for Level 1 Privacy (L1P)

The protocol ensuring level 1 privacy is suitable for users who decide to make friends with each other simply based on the common communities of their overall community

[2]Without loss of generality, we use masculine pronouns for an Initiator and feminine pronouns for a Responder.

Initiator

OFFLINE:

1. Construct polynomial P based on his input set $\overline{C}_I$.
   $P(z) = \prod_{i=1}^{|\overline{C}_I|}(\overline{C}_I^i - z) = \sum_{k=0}^{|\overline{C}_I|} u_k z^k$
2. $\forall i : 0 \leq i \leq |\overline{C}_I|$, Compute $\mathrm{ENC_I}(u_i)$

ONLINE:

1. $\forall i : |0 \leq i \leq |\overline{C}_I|$, send the encrypted coefficients $\mathrm{ENC_I}(u_i)$ along with his public key to the Responder.
2. Choose a random nonce K as key to symmetric encryption function $E_K(\cdot)$ and send $ENC_R(K)$ to the Responder.
3. $\forall i : |1 \leq i \leq |\overline{C}_R|$, send $E_K\big(\mathrm{DEC_I}(\mathrm{ENC_I}(P(\overline{C}_R^i) + R_i))\big) = E_K\big(P(\overline{C}_R^i) + R_i\big)$ to the Responder.

Figure 2.2

Protocol Descriptions (Initiator) of Level 1 Privacy (L1P)

Responder

OFFLINE

1. Assign a random number R_i to all $\overline{C}_R^i \in \overline{C}_R$.

ONLINE:

1. Compute $\mathrm{ENC_I}(P(z)) = \prod_{j=0}^{|\overline{C}_I|} \mathrm{ENC_I}(u_j)^{z^j}$. Send $\mathrm{ENC_I}(P(\overline{C}_R^i) + R_i)$ $(1 \leq i \leq |\overline{C}_R|)$ along with her public key to the Initiator.
2. Acknowledge the receipt by sending $ENC_I(K+1)$ to the Initiator.
3. Find the shared communities by matching R_i (when $P(\overline{C}_R^i) = 0$) with $\overline{C}_R^i$. If the Responder agrees, she accepts the request and sends the encrypted common communities to the Initiator, i.e., $E_K(\overline{C}_I \cap \overline{C}_R)$. Otherwise, the Responder denies the request.

Figure 2.3

Protocol Descriptions (Responder) of Level 1 Privacy (L1P)

sets. In this protocol, we first let the Responder learn the mutual communities and the size of the Initiator's input set ($\overline{C}_I$) (i.e., the Initiator's overall community set), while let the Initiator learn nothing but the size of the Responder's input set ($\overline{C}_R$) (i.e., the Responder's overall community set). Then, the Responder securely sends the common communities to the Initiator, if she confirms the request from the Initiator.

2.4.1.1 Protocol Details

We divide the protocol in two stages: offline and online. In order to speed up the matching process, the Initiator executes part of the protocol offline. In particular, the Initiator uses his input set $\overline{C}_I$ to construct the following polynomial:

$$P(z) = (\overline{C}_I^1 - z)(\overline{C}_I^2 - z).....(\overline{C}_I^{|\overline{C}_I|} - z) = \sum_{k=0}^{|\overline{C}_I|} u_k z^k \tag{2.3}$$

where $\overline{C}_I^i \in \overline{C}_I$ ($1 \leq i \leq |\overline{C}_I|$). He then encrypts the coefficients u_k's of the polynomial and obtains $\mathrm{ENC_I}(u_0), \mathrm{ENC_I}(u_1), ..., \mathrm{ENC_I}(u_{|\overline{C}_I|})$, where $\mathrm{ENC_I}(\cdot)$ is the Initiator's homomorphic encryption function.

As shown in Figure 2.2, in the online stage, the Initiator first sends the encrypted coefficients along with his public key to the Responder. The Responder subsequently constructs (Figure 2.3) the encrypted polynomial based on the encrypted coefficients utilizing the homomorphic property, i.e.,

$$\begin{aligned} \mathrm{ENC_I}(P(z)) &= \mathrm{ENC_I}(u_0)^{z^0} \cdot \mathrm{ENC_I}(u_1)^{z^1} \cdot \\ &\quad .... \cdot \mathrm{ENC_I}(u_{|\overline{C}_I|})^{z^{|\overline{C}_I|}}. \end{aligned} \tag{2.4}$$

The Responder then evaluates $\mathrm{ENC_I}(P(z))$ at each of her own input element, computes the following function, and sends it along with her public key to the Initiator:

$$\mathrm{ENC_I}(P(\overline{C}_R^i) + R_i) = \mathrm{ENC_I}(P(\overline{C}_R^i)) \cdot \mathrm{ENC_I}(R_i) \tag{2.5}$$

where R_i is a random ID generated by the Responder for the community corresponding to $\overline{C}_R^i$, and of the same length as $P(\overline{C}_R^i)$. Then, in the second step, the two parties engage in a challenge response protocol to establish a shared secret key. In particular, the Initiator chooses a random nonce K as the key for a predefined symmetric encryption function $E(\cdot)$ (e.g., AES), encrypts it with the Responder's public key, and sends $ENC_R(K)$ to the Responder, where $ENC_R(\cdot)$ is the Responder's homomorphic encryption function. The Responder recovers K and acknowledges with $ENC_I(K+1)$ to the Initiator. Both parties use K as the shared secret key in the third step. Finally, in the third step, the Initiator decrypts the data received from the Responder in the first step, i.e., $\mathrm{ENC_I}(P(\overline{C}_R^i) + R_i)$'s, encrypts the decrypted data with the symmetric key K using the symmetric encryption algorithm $E_K(\cdot)$, and then sends $E_K\big(DEC_I(ENC_I(\overline{C}_R^i) + R_i))\big) = E_K\big(P(\overline{C}_R^i) + R_i\big)$ back to the Responder. Note that $P(\overline{C}_R^i) = 0$ when the corresponding community $\overline{C}_R^i$ is a mutual community between the Initiator's and the Responder's overall community sets. Thus, after recovering $P(\overline{C}_R^i) + R_i$, the Responder can know the mutual communities by checking R_i's. If she does not want to make friends with the Initiator, she can either ignore or decline the request. Otherwise, she encrypts the mutual communities with the shared secret K and sends $E_K(\overline{C}_I \cap \overline{C}_R)$ to the Initiator, who can now find the shared communities. If he would like to continue, he can finally become friends with the Responder. Note that

to prevent some Initiators from possibly knowing some of the Responder's communities by colluding with each other, the Responder generates a new R_i corresponding to $\overline{C}_R^i$ upon each friendship request.

2.4.1.2 Protocol Analysis

In the following, we analyze the privacy of, and the communication cost and computation cost of the protocol.

Following theorem presents the **Privacy Analysis** of the L1P protocol.

Theorem 1

Before they become friends, the Initiator only learns $|\overline{C}_R|$, and $\overline{C}_I \cap \overline{C}_R$ if the Responder confirms his request, while the Responder only learns $|\overline{C}_I|$ and $\overline{C}_I \cap \overline{C}_R$.

Proof: The Initiator uses semantically secure homomorphic encryption to encrypt the coefficients of the polynomial P, whose roots are the elements of his input set $\overline{C}_I$. The Responder cannot decrypt or distinguish the coefficients, and hence cannot know $\overline{C}_I$ but can learn $|\overline{C}_I|$. Following the protocol, the Responder then sends $\mathrm{ENC_I}(P(\overline{C}_R^i) + R_i)$'s back to the Initiator, where R_i's are random numbers of the same length as $P(\overline{C}_R^i)$'s. Thus, the Initiator can only learn $|\overline{C}_R|$ but nothing more. After receiving $P(\overline{C}_R^i) + R_i$ from the Initiator, the Responder will be able to figure out $\overline{C}_I \cap \overline{C}_R$, and let the Initiator know as well if she decides to confirm the request. Otherwise, the protocol terminates and both parties do not know anything further about each other.

■

The total **Computation and Communication Costs** in this protocol can be analyzed similar to those in [28]. Differently, in the proposed L1P, the Initiator executes part of the protocol offline which in turn reduces the online computation cost. Specifically, the Initiator, in the offline stage, computes the polynomial $P(z)$ and encrypts its coefficients with his public key. As the computational complexity of the exponentiation operation dominates the other operations like multiplication and addition, we analyze the computation overhead focusing on exponentiation operations. Recall that the input set size of the Initiator and of the Responder are $|\overline{C}_I|$ and $|\overline{C}_R|$ respectively, the offline computation cost of the Initiator is $O(|\overline{C}_I|)$ exponentiations. In the online stage, the Initiator's computation cost is $O(|\overline{C}_R|)$ due to decrypting $\mathrm{ENC_I}(P(\overline{C}_R^i) + R_i)$'s received from the Responder. The Responder's computation cost for constructing the encrypted polynomial and evaluating at each of her inputs is $O(|\overline{C}_R| \log\log\ |\overline{C}_I|)$ exponentiations, considering that the polynomial can be efficiently evaluated by Hornor's rule and the balanced bucket allocation scheme presented in [28].

Regarding the communication cost, the Initiator first transmits $O(|\overline{C}_I|)$ encrypted coefficients and the Responder returns $O(|\overline{C}_R|)$ ciphertexts to the Initiator. Subsequently, in the next step, the Initiator returns $O(|\overline{C}_R|)$ decrypted messages and the Responder returns $O(|\overline{C}_I \cap \overline{C}_R|)$ common communities. Thus, the total communication cost for the Initiator is $O(|\overline{C}_I| + |\overline{C}_R|)$ and that for the Responder is $O(|\overline{C}_R| + |\overline{C}_I \cap \overline{C}_R|)$.

Moreover, the L1P protocol allows parallel processing in communication and computation which can further reduce the online execution time. In particular, the Responder does not have to wait for all the coefficients before beginning the computation of the encrypted

polynomial. Similarly, when she starts returning the evaluated encrypted polynomial at each of her input, the Initiator can start decrypting the ciphertexts as soon as he receives one. Hence, if the communication cost is equal to or greater than the online computation overhead in time, the total communication cost would approximately be the total execution time of the protocol.

2.4.2 Protocol for Level 2 Privacy (L2P)

In the protocol for level 1 privacy (L1P), the Responder determines whether or not to accept the Initiator's request for a social friendship only based on their common overall communities, which may not characterize the social proximity well. In this section, we design a protocol for level 2 privacy, called L2P, utilizing the proposed community based asymmetric social proximity measurement. This protocol is suitable for the case when the Initiator is willing to establish a friendship relation with the Responder but the Responder accepts the relationship only if her requirement on the friendship is fulfilled. In particular, in L2P, the Responder accepts the friendship request from the Initiator if the social proximity measured by her, i.e, $\Psi_{R\leftarrow I}$, is greater than a threshold predefined by herself, denoted by Ψ_{R_τ}. The protocol is detailed as follows.

2.4.2.1 Protocol Details

Similar to that in L1P, an Initiator and a Responder can execute part of the protocol offline in order to speed up the matching process.

In the **OFFLINE** phase, the same as that in L1P, the Initiator constructs the polynomial P, with his inputs $\overline{C}_I = \{\overline{C}_I^1, \overline{C}_I^2, ...\overline{C}_I^{|\overline{C}_I|}\}$ being the roots, and encrypts the coefficients

using his own homomorphic encryption function $\mathrm{ENC_I}(\cdot)$. On the other hand, the Responder calculates the partial social proximity corresponding to each of her overall communities as follows:

$$\Psi^i_{R\leftarrow I} = \frac{\sum_{j\in\overline{C}^i_R\cap\overline{\mathcal{N}}_R}\left(\beta_j(\overline{C}^i_R)\sum_{\{k|k\in FC(R,j)\}}\left(\alpha^k_R\right)\right)}{\sum_{i=1}^{|\overline{C}_R|}\sum_{j\in\overline{C}^i_R\cap\overline{\mathcal{N}}_R}\left(\beta_j(\overline{C}^i_R)\sum_{\{k|k\in FC(R,j)\}}\left(\alpha^k_R\right)\right)\left(\right.} \tag{2.6}$$

for any $\overline{C}^i_R \in \overline{C}_R$ ($1 \le i \le |\overline{C}_R|$). The Responder needs to encrypt the partial social proximity for all $\overline{C}^i_R$'s with her public key. However, $\Psi^i_{R\leftarrow I}$ is a fractional number and general additive homomorphic schemes cannot be used to encrypt the fractional numbers. Note that since α's and β's are integers, $\Psi^i_{R\leftarrow I}$ is a rational number. Besides, the denominator in (2.6) is a constant for all $\overline{C}^i_R \in \overline{C}_R$. We denote the denominator by D_R. The Responder encrypts the numerator (integer) of the partial social proximity with her public key, i.e., computes $\mathrm{ENC_R}(\Psi^i_{R\leftarrow I} \cdot D_R)$, where $\mathrm{ENC_R}(\cdot)$ is her homomorphic encryption function. In addition, the Responder assigns a random ID, R_i, to each of her overall communities upon each friendship request.

In the **ONLINE** phase, i.e., when an Initiator and a Responder decide to execute the protocol, the Initiator first sends the encrypted coefficients of the polynomial P to the Responder. Note that the Initiator and the Responder exchange their public keys to establish a shared secret key K in the same way as that in L1P, which is also shown in Figure 2.4, Figure 2.5. The detailed description of shared key establishment is omitted below to avoid redundancy. The Responder then constructs the encrypted polynomial according to (2.4), and evaluates the polynomial at each of her input $\overline{C}^i_R \in \overline{C}_R$. Taking advantage of the

homomorphic property of the encryption, the Responder further constructs the following message

$$(A_i, B_i, C_i) = \Big(\mathrm{ENC_I}(\rho_i \cdot P(\overline{C}_R^i)), \; \mathrm{ENC_I}\left(P(\overline{C}_R^i) + \mathrm{ENC_R}(\Psi_{R\leftarrow I}^i \cdot D_R)\right) \Big(R_i \Big) \tag{2.7}$$

for each $\overline{C}_R^i \in \overline{C}_R$, where ρ_i is a random number of the same length as $P(y_i)$, and sends (A_i, B_i, C_i) to the Initiator.

The Initiator then decrypts A_i, and for each i with $\mathrm{DEC_I}(A_i) = 0$, calculates $\mathrm{DEC_I}(B_i) = (0 + \mathrm{ENC_R}(\Psi_{R\leftarrow I}^i) \cdot D_R) = \mathrm{ENC_R}(\Psi_{R\leftarrow I}^i \cdot D_R)$, which implies the corresponding input $\overline{C}_R^i \in (\overline{C}_I \cap \overline{C}_R)$. After that, the Initiator can calculate the encrypted social proximity for the Responder by aggregating all B_i's as follows:

$$\begin{aligned}
&\mathrm{ENC_R}(\Psi_{R\leftarrow I} \cdot D) \\
&= \prod_{\{i|\mathrm{DEC_I}(A_i)=0\}} \Big(\mathrm{DEC_I}(B_i) \\
&= \prod_{\{i|\mathrm{DEC_I}(A_i)=0\}} \Big(\mathrm{ENC_R}\Big(\sum_{j\in\overline{C}_R^i\cap\overline{\mathcal{N}}_R} \Big(\beta_j(\overline{C}_R^i) \cdot \Big(\sum_{\{k|k\in FC(R,j)\}} \alpha_R^k \Big) \Big) \Big) \Big) \\
&= \mathrm{ENC_R}\Big(\sum_{i=1}^{|C_{IR}|} \sum_{j\in\overline{C}_R^i\cap\overline{\mathcal{N}}_R} \Big(\beta_j(\overline{C}_R^i) \sum_{\{k|k\in FC(R,j)\}} \Big(\alpha_R^k \Big) \Big) \Big) \\
&= \mathrm{ENC_R}(\Psi_{R\leftarrow I} \cdot D_R)
\end{aligned} \tag{2.8}$$

Obviously, we can see that when $A_i = 0$, the term B_i gives the numerator of the encrypted (by the Responder) partial social proximity attributed to the community C_R^i that is common to both the Initiator and the Responder. Thus, due to homomorphic property, the product

Initiator

OFFLINE:

1. Construct a polynomial P based on his input set $\overline{C}_I$. i.e., $P(z) = \prod_{i=1}^{|\overline{C}_I|}(\overline{C}_I^i - z) = \sum_{k=0}^{|\overline{C}_I|} u_k z^k$
2. $\forall i : 0 \leq i \leq |\overline{C}_I|$, compute $\mathrm{ENC_I}(u_i)$

ONLINE:

1. $\forall i : 0 \leq i \leq |\overline{C}_I|$, send the encrypted coefficients $\mathrm{ENC_I}(u_i)$ along with his public key to the Responder.
2. Decrypt (A_i, B_i)'s for $1 \leq i \leq |\overline{C}_R|$. Compute $\mathrm{ENC_R}(\Psi_{R\leftarrow I} \cdot D_R)$ according to (2.7). Choose three large random numbers r_1, r_2, r_3, where $0 \ll r_1 \leq r_2 \leq r_3$ and $\epsilon < (r_2 - r_1)/r_3 \ll \Psi^{min}$. Compute and send the tuple $(M, N) = (\mathrm{ENC_R}(r_1 + r_3(\Psi_{R\leftarrow I} \cdot D_R)), \mathrm{ENC_R}(r_3(\Psi_{R_\tau} \cdot D_R) + r_1))$ to the Responder. Besides, choose a random nonce K as a key to symmetric encryption function $E(\cdot)$ and send $ENC_R(K)$ along with the tuple.
3. Encrypt the corresponding ID (R_i) for any $\mathrm{DEC_I}(A_i) = 0$ as $E_K(R_i)$, and send to the Responder.

Figure 2.4

Protocol Descriptions (Initiator) of Extended Level 2 Privacy (EL2P).

Responder

OFFLINE

1. Assign a random R_i to each $\overline{C}_R^i \in \overline{C}_R$.
2. $\forall i : 0 \leq i \leq |\overline{C}_R|$, compute $\mathrm{ENC_R}(\Psi_{R \leftarrow I}^i \cdot D_R)$ based on (2.5).

ONLINE:

1. Compute $\mathrm{ENC_I}(P(z)) = \prod_{j=0}^{|\overline{C}_I|} \mathrm{ENC_I}(u_j)^{z^j}$. Send $(A_i, B_i, C_i) = \big(\mathrm{ENC_I}(\rho_i \cdot P(\overline{C}_R^i)), \mathrm{ENC_I}(P(\overline{C}_R^i) + \mathrm{ENC_R}(\Psi_{R \leftarrow I}^i \cdot D_R)), R_i\big) (1 \leq i \leq |\overline{C}_R|)$, and $\mathrm{ENC_R}(\Psi_{R_\tau} \cdot D_R)$ along with her public key to the Initiator.
2. Check if $\mathrm{DEC_R}(M) \geq \mathrm{DEC_R}(N)$, i.e. $\Psi_{R \leftarrow I} > \Psi_{R_\tau}$. If so, decrypt $ENC_R(K)$ and acknowledge the reciept by sending $ENC_I(K+1)$. Otherwise, the Responder denies the request and aborts.
3. Decrypt the received data to obtain mutual communities $\overline{C}_I \cap \overline{C}_R$ corresponding to the received R_i's, and send $E_K(\overline{C}_I \cap \overline{C}_R)$ to the Initiator.

Figure 2.5

Protocol Descriptions (Responder) of Extended Level 2 Privacy (EL2P).

of the encryption over all the communities with $A_i = 0$ is equal to the encryption of the sum of the partial social proximities attributed to all the common communities shared by the Initiator and the Responder, as shown in (2.8). As is evident from (2.2), (2.8) is in fact the encrypted (by the Responder) social proximity between the Responder and the Initiator gauged by the Responder times D_R, i.e., $ENC_R(\Psi_{R\leftarrow I} \cdot D_R)$.

The Initiator sends $\mathrm{ENC_R}(\Psi_{R\leftarrow I} \cdot D_R)$ to the Responder, who decrypts it and checks to see if $\Psi_{R\leftarrow I} \cdot D_R \geq \Psi_{R_\tau} \cdot D_R$, i.e. $\Psi_{R\leftarrow I} \geq \Psi_{R_\tau}$. If not, the Responder aborts the protocol and informs the Initiator. Otherwise, the Responder confirms the Initiator's request, who then encrypts the R_i's, corresponding to the cases when $\mathrm{DEC_I}(A_i) = 0$, with the symmetric key K and sends $E_K(R_i)$'s to the Responder. After decrypting $E_K(R_i)$, the Responder can then know $\overline{C}_I \cap \overline{C}_R$ and sends $E_K(\overline{C}_I \cap \overline{C}_R)$ back to the Initiator, who can now become friends with the Responder if he still would like to proceed.

2.4.2.2 Extended Protocol for Level 2 Privacy (EL2P)

In the above L2P protocol, it is possible that the Responder may learn more than just the social proximity $\Psi_{R\leftarrow I}$ when receiving $\Psi_{R\leftarrow I} \cdot D_R$ and hence $\Psi_{R\leftarrow I}$ (the Responder knows D_R) from the Initiator. For example, if there happens to be only one common community between the Initiator and the Responder, then it is possible for the Responder to find out the common community by looking at the partial social proximity $\Psi^i_{R\leftarrow I}$ value of each of her communities even if $\Psi_{R\leftarrow I} \ngeq \Psi_{R_\tau}$. Similarly, even if there are multiple common communities shared by the Initiator and the Responder, the Responder may learn the common communities by checking if the sum of several partial social proximity is

equal to $\Psi_{R\leftarrow I}$ received from the Initiator. Here, we extend the L2P protocol so that the Responder only learns whether $\Psi_{R\leftarrow I} \geq \Psi_{R_\tau} + \epsilon/D_R$, where ϵ is a small number such that $\epsilon \ll \Psi_{R_\tau}$, instead of the value of $\Psi_{R\leftarrow I}$. The detailed process of EL2P is described in Figure 2.4 for the Initiator and Figure 2.5 for the Responder.

Specifically, at the end of step 1) of the online phase, the Responder sends $\mathrm{ENC_R}(\Psi_{R_\tau} \cdot D_R)$ in addition to (A_i, B_i, C_i) to the Initiator. The Initiator then computes $\mathrm{ENC_R}(\Psi_{R\leftarrow I} \cdot D_R)$ according to (2.8), and chooses three large positive random numbers r_1, r_2, and r_3 such that $0 \ll r_1 < r_2 < r_3$ and $\epsilon < \frac{r_2 - r_1}{r_3} \ll \Psi^{min}$, where Ψ^{min} is a predefined minimum social proximity threshold and known to all the users. Note that

$$\begin{aligned} & (r_1 + r_3(\Psi_{R\leftarrow I} \cdot D_R)) \geq (r_3(\Psi_{R_\tau} \cdot D_R) + r_2) \\ \Longrightarrow \quad & \Psi_{R\leftarrow I} \geq \left(\Psi_{R_\tau} + \frac{(r_2 - r_1)/r_3}{D_R}\right)\Bigg(\\ \Longrightarrow \quad & \Psi_{R\leftarrow I} \geq \Psi_{R_\tau} + \epsilon/D_R \end{aligned} \tag{2.9}$$

Therefore, the Initiator can compute $(M, N) = \Big(\mathrm{ENC_R}(r_1 + r_3(\Psi_{R\leftarrow I} \cdot D_R)), \mathrm{ENC_R}(r_3(\Psi_{R_\tau} \cdot D_R) + r_2)\Big)\Bigg($ as follows

$$\begin{aligned} & \mathrm{ENC_R}(r_1 + r_3(\Psi_{R\leftarrow I} \cdot D_R)) \\ = \quad & \mathrm{ENC_R}(r_1) \cdot \mathrm{ENC_R}(\Psi_{R\leftarrow I} \cdot D_R)^{r_3} \\ & \mathrm{ENC_R}(r_3(\Psi_{R_\tau} \cdot D_R) + r_2) \\ = \quad & \mathrm{ENC_R}(\Psi_{R_\tau} \cdot D_R)^{r_3} \cdot \mathrm{ENC_R}(r_2) \end{aligned} \tag{2.10}$$

and sends (M, N) back to the Responder (instead of sending $\mathrm{ENC_R}(\Psi_{R\leftarrow I} \cdot D_R)$ to her). The Responder then checks to see if $\mathrm{DEC_R}(M) \geq \mathrm{DEC_R}(N)$ and follows the rest of the protocol accordingly in the same way as presented above.

We can see that in this extended protocol EL2P, the Responder is only able to learn if $\Psi_{R \leftarrow I} \geq \Psi_{R_\tau} + \epsilon/D_R$, i.e., $\Psi_{R \leftarrow I} > \Psi_{R_\tau}$ (since $\epsilon/D_R \ll \Psi^{min}/D_R \ll \Psi_{R_\tau}$), and the above problem can be addressed.

2.4.2.3 Protocol Analysis

In the following we analyze the EL2P protocol in terms of privacy, and computation and communication cost.

The following theorem presents **Privacy Analysis** of EL2P protocol.

Theorem 2

Before they become friends, the Initiator learns only $|\overline{C}_R|$ and $|\overline{C}_I \cap \overline{C}_R|$, and the mutual communities $\overline{C}_I \cap \overline{C}_R$ if $\Psi_{R \leftarrow I} > \Psi_{R_\tau}$, while the Responder learns only $|\overline{C}_I|$, and the mutual communities $\overline{C}_I \cap \overline{C}_R$ if $\Psi_{R \leftarrow I} > \Psi_{R_\tau}$.

Proof: The Initiator uses semantically secure homomorphic encryption to encrypt the coefficients of the polynomial P, whose roots are the elements of his input set $\overline{C}_I$. The Responder cannot decrypt or distinguish the coefficients, and hence cannot know $\overline{C}_I$ but can learn $|\overline{C}_I|$. Following the protocol, the Responder then sends (A_i, B_i, C_i)'s to the Initiator, who can then know $|\overline{C}_R|$. By decrypting (A_i, B_i, C_i)'s and counting all A_i's that are decrypted to be 0, the Initiator can then know the size of the mutual community set, i.e., $|\overline{C}_I \cap \overline{C}_R|$, but does not know which the mutual communities are. He then computes the tuple (M, N), and sends it to the Responder. If the Responder finds $\Psi_{R \leftarrow I} > \Psi_{R_\tau}$, she informs the Initiator who sends her the random IDs R_i's, and hence can know the mutual communities $\overline{C}_I \cap \overline{C}_R$. Otherwise, the protocol terminates and both parties do not know

anything further about each other. Besides, similarly to that in Theorem 1, the Initiator and the Responder cannot know all the communities in each other's overall community set by artificially extending their input sets.

Moreover, one may argue that it is possible for the Initiator to cheat by increasing M, for example, computing $M' = M \cdot \mathrm{ENC}_{\mathrm{R}}(r_4)$ where $r_4 > 0$ or $M' = M^{r_4}$ where $r_4 > 1$, so that the Responder will get $\mathrm{DEC}_{\mathrm{R}}(M) > \mathrm{DEC}_{\mathrm{R}}(N)$ and hence accepts his request. However, the Initiator will always be caught since the Responder can verify in step 3) of the online phase whether or not $\Psi_{R \leftarrow I} > \Psi_{R_\tau}$ by checking the received R_i's from the Initiator before revealing $\overline{C}_I \cap \overline{C}_R$ to the Initiator. Without receiving the mutual communities, the Initiator cannot finally be authorized to make friends with the Responder.

■

The **Computation and Communication Costs** of the protocol are calculated as follows. The Initiator and the Responder execute part of the protocol offline, as in L1P, which can reduce the online computation time. In particular, in the offline phase, the Initiator incurs $O(|\overline{C}_I|)$ exponentiations to compute the encrypted coefficients of $P(z)$. Similarly, the Responder has a computation load of $O(|\overline{C}_R|)$ exponentiations to compute the partial social proximity offline. In the online phase, the computation cost for the Initiator is $O(|\overline{C}_R|)$ exponentiations (in step 2) of the online phase as shown in Figure 2.4). The Responder performs $O(|\overline{C}_R| \log\log |\overline{C}_I|)$ exponentiations (Figure 2.5) in step 1) of the online phase.

As for the communication cost, the Initiator sends $O(|\overline{C}_I|)$ encrypted coefficients in step 1) and $O(|\overline{C}_I \cap \overline{C}_R|)$ IDs (R_i's) in step 3) of the protocol. The Responder, on the other

hand, replies with $O(|\overline{C}_R|)$ encrypted ciphertexts in step 1) and $O(|\overline{C}_I \cap \overline{C}_R|)$ communities in step 3). Thus, the total communication overhead for the Initiator is $O(|\overline{C}_I| + |\overline{C}_I \cap \overline{C}_R|)$ and that for the Responder is $O(|\overline{C}_R| + |\overline{C}_I \cap \overline{C}_R|)$.

2.4.3 Protocol for Level 3 Privacy

In the L2P protocol, the Responder determines whether or not to be friends with the Initiator based on the community based social proximity, while the Initiator still can only make his final decision based on their common communities. Besides, in terms of privacy, in L2P the Responder will know $\overline{C}_I \cap \overline{C}_R$ if $\Psi_{R \leftarrow I} > \Psi_{R_\tau}$, no matter whether the social proximity measured by the Initiator is large enough or not. In this section, we develop a protocol for level 3 privacy, called L3P, to address the above problems. This protocol is suitable for users with very high privacy requirements. In this protocol, both the Initiator and the Responder make sure their requirements on friendship are fulfilled before revealing any matching information to each other. If either of the requirements is not satisfied, neither of them knows the matching profile information, i.e., the common communities $\overline{C}_I \cap \overline{C}_R$.

2.4.3.1 Protocol Description

The same as that in L1P and L2P, part of the L3P protocol can be completed offline. In what follows, we briefly describe the offline and online phases of the protocol, respectively, which are also shown in Figure 2.6 for the Initiator and Figure 2.7 for the Responder.

In the **OFFLINE** phase, the Initiator constructs a polynomial P with his input set $\overline{C}_I$ being the roots, while the Responder constructs a polynomial Q with her input set $\overline{C}_R$ being the roots (step 1)). Each of them encrypts the coefficients of their polynomials using

Initiator

OFFLINE:

1. Construct polynomial P based on his input set $\overline{C}_I$.
   $P(z) = \prod_{i=1}^{\overline{C}_I}(\overline{C}_I^i - z) = \sum_{k=0}^{|\overline{C}_I|} u_k z^k$
2. $\forall i : 0 \leq i \leq |\overline{C}_I|$, compute $\mathrm{ENC_I}(u_i)$
3. $\forall i : 0 \leq i \leq |\overline{C}_I|$, compute $\mathrm{ENC_I}(\Psi^i_{I \leftarrow R} \cdot D_I) =$
   $$\mathrm{ENC_I}\left(\sum_{j \in \overline{C}_I^i \cap \overline{\mathcal{N}}_I}\left(\beta_j(\overline{C}_I^i)\sum_{\{k|k \in FC(I,j)\}}\alpha_I^k\right)\right)$$

ONLINE:

1. $\forall i : 0 \leq i \leq |\overline{C}_I|$, send the encrypted coefficients $\mathrm{ENC_I}(u_i)$ to the Responder.
2. Compute $\mathrm{ENC_R}(Q(z)) = \prod_{j=0}^{|\overline{C}_R|}\mathrm{ENC_R}(v_j)^{z^j}$. Send $(A'_i, B'_i) = \big(\mathrm{ENC_R}(\rho'_i \cdot P(\overline{C}_I^i)), \mathrm{ENC_R}(\rho'_i \cdot P(\overline{C}_I^i) + \mathrm{ENC_I}(\Psi^i_{I \leftarrow R} \cdot D_I))\big)$ $(1 \leq i \leq |\overline{C}_I|)$, and $\mathrm{ENC_I}(\Psi_{I_\tau} \cdot D_I)$ to the Responder.
3. Decrypt (A_i, B_i)'s for $1 \leq i \leq |\overline{C}_R|$. Compute $\mathrm{ENC_R}(\Psi_{R \leftarrow I} \cdot D_R)$ according to (2.7). Choose three large random numbers r'_1, r'_2, r'_3 where $0 \ll r'_1 < r'_2 < r'_3$ and $\epsilon < (r'_2 - r'_1)/r'_3 \ll \Psi_{R_\tau}$. Compute and send the tuple $(M', N') = (\mathrm{ENC_R}(r'_1 + r'_3(\Psi_{R \leftarrow I} \cdot D_R)), \mathrm{ENC_R}(r'_3(\Psi_{R_\tau} \cdot D_R) + r'_1))$ to the Responder.
4. Send the result of the comparision, i.e., whether $\mathrm{DEC_I}(M) \geq \mathrm{DEC_I}(N)$, i.e., $\Psi_{I \leftarrow R} > \Psi_{I_\tau}$ or not.

Figure 2.6

Protocol Descriptions (Initiator) of Level 3 Privacy (L3P).

Responder

OFFLINE:

1. Construct polynomial Q from input set $\overline{C}_R$.
   $Q(z) = \prod_{i=1}^{|\overline{C}_R|}(\overline{C}_R^i - z) = \sum_{k=0}^{|\overline{C}_R|} v_k z^k$
2. $\forall i : 0 \leq i \leq |\overline{C}_R|$, compute $\mathrm{ENC_R}(v_i)$
3. $\forall i : 0 \leq i \leq |\overline{C}_R|$, compute $\mathrm{ENC_R}(\Psi^i_{R\leftarrow I} \cdot D_R) =$
   $$\mathrm{ENC_R}\left(\sum_{j\in\overline{C}_R^i\cap\overline{\mathcal{N}}_R}\left(\beta_j(\overline{C}_R^i)\sum_{\{k|k\in FC(R,j)\}}\alpha_R^k\right)\right)$$

ONLINE:

1. $\forall i : 0 \leq i \leq |\overline{C}_R|$, Send the encrypted coefficients $\mathrm{ENC_R}(v_i)$ to the Responder.
2. Compute $\mathrm{ENC_I}(P(z)) = \prod_{j=0}^{|\overline{C}_I|}\mathrm{ENC_I}(u_j)^{z^j}$. Send $(A_i, B_i) = \big(\mathrm{ENC_I}(\rho_i \cdot P(\overline{C}_R^i)), \mathrm{ENC_I}(\rho_i \cdot P(\overline{C}_R^i) + \mathrm{ENC_R}(\Psi^i_{R\leftarrow I} \cdot D_R))\big)(1 \leq i \leq |\overline{C}_R|)$, and $\mathrm{ENC_R}(\Psi_{R_\tau} \cdot D_R)$ to the Initiator.
3. Decrypt (A'_i, B'_i)'s for $1 \leq i \leq |\overline{C}_R|$. Compute $\mathrm{ENC_I}(\Psi_{I\leftarrow R} \cdot D_I)$ in a similar way to (2.7). Choose three large random numbers r_1, r_2, r_3 where $0 \ll r_1 < r_2 < r_3$ and $\epsilon < (r_2 - r_1)/r_3 \ll \Psi_{I_\tau}$. Compute and send the tuple $(M, N) = (\mathrm{ENC_I}(r_1 + r_3(\Psi_{I\leftarrow R} \cdot D_I)), \mathrm{ENC_I}(r_3(\Psi_{I_\tau} \cdot D_I) + r_1))$ to the Initiator.
4. Send the result of the comparison, i.e., whether $\mathrm{DEC_R}(M') \geq \mathrm{DEC_R}(N')$, i.e., $\Psi_{R\leftarrow I} > \Psi_{R_\tau}$ or not.

Figure 2.7

Protocol Descriptions (Responder) of Level 3 Privacy (L3P).

their own public keys in step 2), and computes partial community based social proximities in step 3).

In the **ONLINE**, the Initiator and the Responder exchange their encrypted coefficients in step 1). The Initiator and the Responder construct $\mathrm{ENC_R}(Q(z))$ and $\mathrm{ENC_I}(P(z))$, respectively, based on the received ciphertexts, and evaluate at each of their own inputs, and exchange their tuples (A'_i, B'_i) and (A_i, B_i), i.e.,

$$\begin{pmatrix} (\mathrm{ENC_R}(\rho'_i \cdot P(\overline{C}^i_I)), \mathrm{ENC_R}(\rho'_i \cdot P(\overline{C}^i_I) + \mathrm{ENC_I}(\Psi^i_{I\leftarrow R} \cdot D_I)) \\ (\mathrm{ENC_I}(\rho_i \cdot P(\overline{C}^i_R)), \mathrm{ENC_I}(\rho_i \cdot P(\overline{C}^i_R) + \mathrm{ENC_R}(\Psi^i_{R\leftarrow I} \cdot D_R)) \end{pmatrix}$$

along with $\mathrm{ENC_I}(\Psi_{I_\tau} \cdot D_I)$ and $\mathrm{ENC_R}(\Psi_{R_\tau} \cdot D_R)$, respectively, in step 2). Note that ρ'_i and ρ_i are random numbers of the same length as $P(\cdot)$, Similar to that in step 2) of the L2P online phase, the Initiator and the Responder exchange the tuples (M', N') and (M, N) in step 3), i.e.,

$$(\mathrm{ENC_R}(r'_1 + r'_3(\Psi_{R\leftarrow I} \cdot D_R)), \mathrm{ENC_R}(r'_3(\Psi_{R_\tau} \cdot D_R) + r'_1))$$

$$(\mathrm{ENC_I}(r_1 + r_3(\Psi_{I\leftarrow R} \cdot D_I)), \mathrm{ENC_I}(r_3(\Psi_{I_\tau} \cdot D_I) + r_1)).$$

where $0 \ll r'_1 < r'_2 < r'_3$ and $\epsilon < \frac{r'_2 - r'_1}{r'_3} \ll \Psi^{min}$, and $0 \ll r_1 < r_2 < r_3$ and $\epsilon < \frac{r_2 - r_1}{r_3} \ll \Psi^{min}$. If at least one of the social proximity criteria is not satisfied, i.e., if $\Psi_{I\leftarrow R} \not> \Psi_{I_\tau}$or/and $\Psi_{R\leftarrow I} \not> \Psi_{R_\tau}$, they cannot become friends and the protocol stops at step 4) before either of them is able to learn any matching information. Otherwise, i.e., if $\Psi_{I\leftarrow R} > \Psi_{I_\tau}$ and $\Psi_{R\leftarrow I} > \Psi_{R_\tau}$ both hold, the Initiator and the Responder are both assumed to be willing to establish a social friendship and they can become friends now.

2.4.3.2 Protocol Analysis

Next, we present the analysis on the privacy, and computation and communication cost of the L3P protocol.

The following theorem presents **Privacy Analysis** of the L3P protocol.

Theorem 3

Before they become friends, the Initiator learns $|\overline{C}_R|$ and $|\overline{C}_I \cap \overline{C}_R|$, while the Responder learns $|\overline{C}_I|$ and $|\overline{C}_I \cap \overline{C}_R|$.

Proof: The proof is similar to that of Theorem 2 and hence omitted here. ■

Moreover, in most previous schemes, e.g., [16, 22, 46, 47, 50, 53, 75], a user can request friendship with another user, run some protocols, and then leave with the user's private information before the user can know anything. In our schemes, as shown in Theorems 1-3, when one user requests friendship with another, he/she can know some of the user's important private information only if the user is willing to accept the request.

The **Computation and Communication Costs** of the protocol is analyzed in the following. A significant fraction of the computation in L3P can be done offline. In particular, as shown in Figure 2.6, the Initiator performs $(|\overline{C}_I| + 1)$ encryptions on the coefficients of the polynomial $P(z)$ and also computes $|\overline{C}_I|$ encryptions on partial social proximity measurements. Thus, the Initiator's total offline computation complexity is $O(|\overline{C}_I|)$ exponentiations. Similarly, the Responder's total offline computation complexity, as shown in Figure 2.7, is $O(|\overline{C}_R|)$ exponentiations. In the online phase, following the similar analysis to that of the previous two protocols, the Initiator's computation complexity in step 2) is $O(|\overline{C}_I| \log\log |\overline{C}_R|)$ exponentiations and that in step 3) is $O(|\overline{C}_R|)$ exponentiations. The

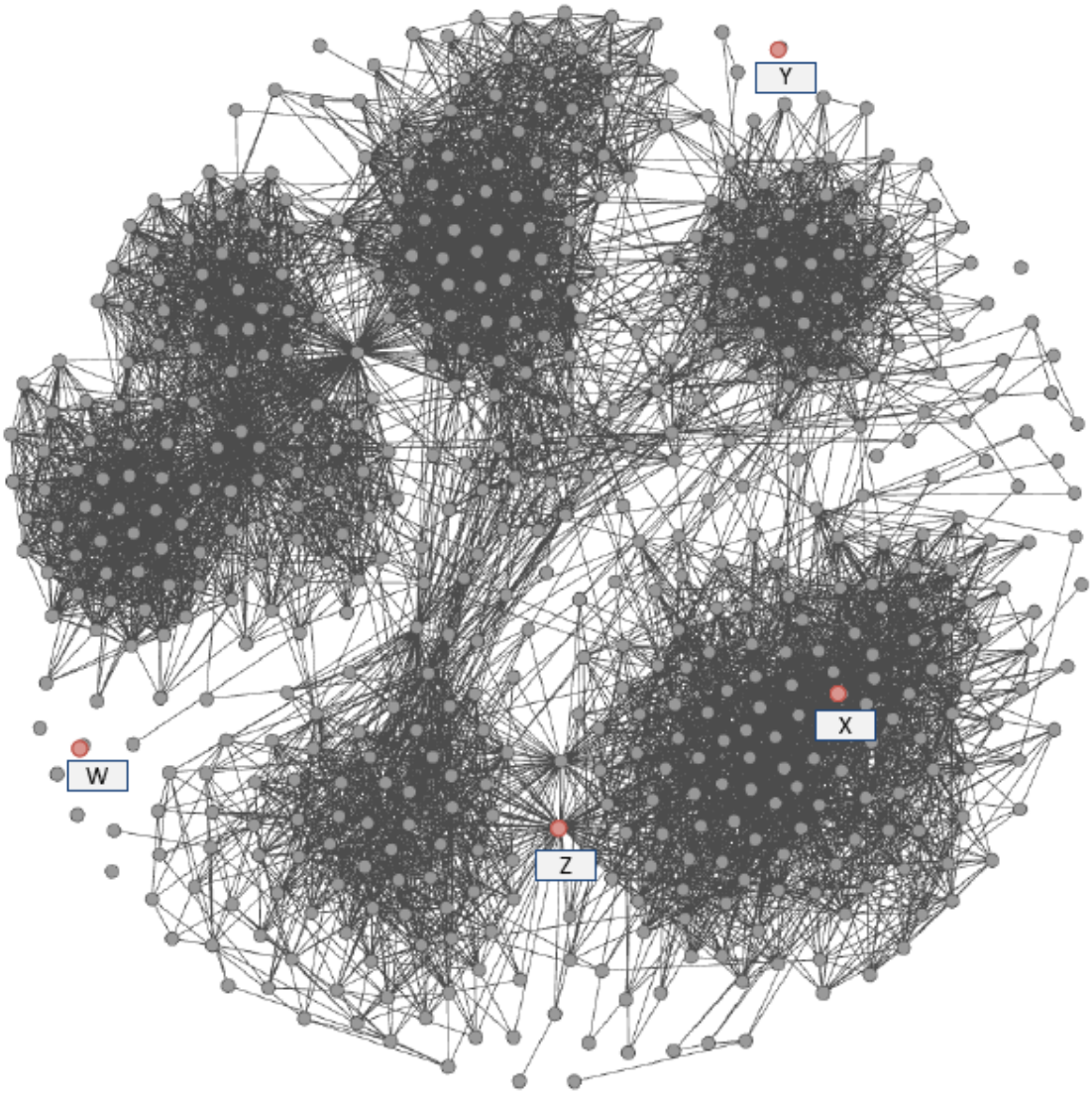

Figure 2.8

Facebook Ego-Network of 'A'

Responder's computation complexity is $O(|\overline{C}_R| \log\log |\overline{C}_I|)$ and $O(|\overline{C}_I|)$ exponentiations in step 2) and step 3), respectively.

Moreover, the Initiator's communication cost is $O(|\overline{C}_I|)$ and $O(|\overline{C}_R|)$ in step 1) and step 2), respectively. Similarly, the Responder's communication cost is $O(|\overline{C}_R|)$ and $O(|\overline{C}_I|)$ in step 1) and step 2), respectively. Therefore, both the Initiator and the Responder have a total communication cost of $O(|\overline{C}_R| + |\overline{C}_I|)$. In addition, as mentioned before, some computation and communication can be done in parallel, thus reducing the overall protocol execution time.

2.5 Performance Evaluation

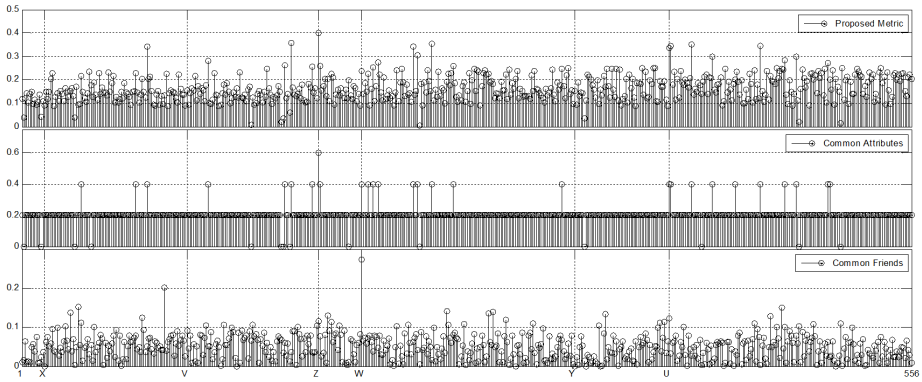

Figure 2.9

Calculated Proximity between A and A's friends using Three different Metrics

2.5.1 Asymmetric Social Proximity Measure Validation

In Section 2.3.2, we propose an asymmetric social proximity metric between two users, which is based on each user's as well as his/her friends' perceptions on the common communities between the two users. In this section, we design an experiment to validate the proposed metric using one author's (whom we denote by A) Facebook ego-network as shown in Figure 2.8. The ego-network has 556 nodes (A's friends) and 7856 edges (interconnections among A's friends). The degree of each node in the network gives the number of common friends between A and the node (A's friend). Note that A is not in the network.

In order to quantify the proximity between A and any of his friends according to the asymmetric proximity metric proposed in this work, A divides his friends into the following six friend circles: $FC_A = \{FC_A^1, FC_A^2, FC_A^3, FC_A^4, FC_A^5, FC_A^6\} = \{$Friends from hometown, Friends in the current university, Friends from the previous university, Job 1 friends, Job 2 friends, Others$\}$[3]. Starting clockwise from the large cluster in the lower right in Figure 2.8, the clusters correspond to FC_A^1 to FC_A^5 respectively. We look at each node (A's friend) in the network, and associate it with one or more communities according to its current and previous locations, occupations, academic institutions, etc. For example, a node V in the network can be a member of city T_i and city T_j communities, the ECE department of university U_i community, and the organization O_i community. The values of α and β are set from 0 to 10. In this experiment, A assigns

[3]Interestingly, the different clusters in the ego-network as shown in Fig. 2.8 approximately represent these different friend circles of A (except "Others"). This opens up the possibility of automating the process of dividing one's friends into different friend circles. The weights on the friend circles can be estimated automatically, e.g., based on the number of friends in them, and finally confirmed by the user. The weights on the communities can be estimated similarly.

$\alpha_A^1 = 10, \alpha_A^2 = 9, \alpha_A^3 = 9, \alpha_A^4 = 7, \alpha_A^5 = 5$, and $\alpha_A^6 = 2$. Besides, for all $V \in \overline{\mathcal{N}}_A$ and $C_V^j \in C_V$, we have $\beta_V(C_V^j) = 10$ if $|C_V^j| \geq 50$, and $\beta_V(C_V^j) = 5$ otherwise.

Figure 2.5 shows the social proximity values computed based on the normalized number of common friends, the normalized number of common profile attributes, and the asymmetric proximity metric proposed in this work for each of the 556 nodes (friends of A) in the network. In particular, the normalized number of common friends is calculated as the number of common friends between A and one of A's friends divided by the total number of possible common friends, i.e., 556, in this case. Similarly, the number of common profile attributes (i.e., communities here) is normalized regarding the total number of profile attributes (communities) of A, i.e., $|C_A|$. We contend that compared to the other two metrics, the proposed asymmetric proximity measure can better describe the friendship valuations. In the following, we choose four nodes (W, X, Y, Z) to compare these three metrics in detail.

Specifically, the normalized number of common friends cannot fully differentiate the importance of friends. For example, Z and X share approximately the same number of friends with A, and their normalized numbers of common friends with A are 0.11 and 0.13, respectively. In contrast, the proposed asymmetric proximity of Z is nearly twice as much as that of X ($\Psi_{A \leftarrow Z} = 0.39, \Psi_{A \leftarrow X} = 0.20$), since Z shares two communities with A and belongs to two different friend circles FC_A^1, FC_A^2 while X only shares one community with A and belongs to only one friend circle FC_A^1 . The higher social proximity value of Z is justified from the network theory perspective. Particularly, the ratio of *betweenness*

centrality[4] of Z to that of X is 5.5 : 1, which emphasizes the relative importance of node Z over X.

Similarly, the normalized number of common attributes fails to well differentiate the importance of friends as well. In our experiment, as shown in Figure 2.5, most nodes have the same normalized number of common attributes, and hence cannot be differentiated based on this metric. More importantly, it also fails to fully establish friendships whenever possible. For example, many of A's friends do not have any common attributes with A and hence their normalized numbers of common attributes are 0. On the other hand, the proposed asymmetric proximity measure gives non-zero values as those friends share attributes with some other friends of A. The experiment confirms our argument in the beginning that whether two people can become friends not only depends on whether they have anything (attributes) in common, but also depends on whether their friends have anything in common.

To give another example, friends W and Y have the same normalized number of common friends (0.2) and the same number of common attributes (0). In contrast, the proposed asymmetric proximity measure is able to differentiate these two friends, i.e., 0.18 and 0.11, respectively, as they are associated with different communities and belong to different friend circles with different sizes and weights.

Moreover, we conduct similar experiments on ego-networks of Z, and find that $\Psi_{Z \leftarrow A} = 0.46$ which is larger than $\Psi_{A \leftarrow Z}$, i.e., 0.39, as shown above. Apparently, Z values friend-

[4]Betweenness centrality is a measure of a node's centrality in a network [29]. The betweenness centrality of a node v in a network is equal to the number of shortest paths from all nodes to all others that pass through node v.

ship with A more than A does to friendship with Z. This is because the size overall community set of A is about 15 percent larger than that of Z ($|\overline{C}_A| = 192, |\overline{C}_Z| = 165$). Besides, A shares two communities with Z and is in to two of the total four different friend circles of Z, whereas Z is in two of the six friend circles of A. This demonstrates the asymmetric characteristics of friendships captured by our proximity measure.

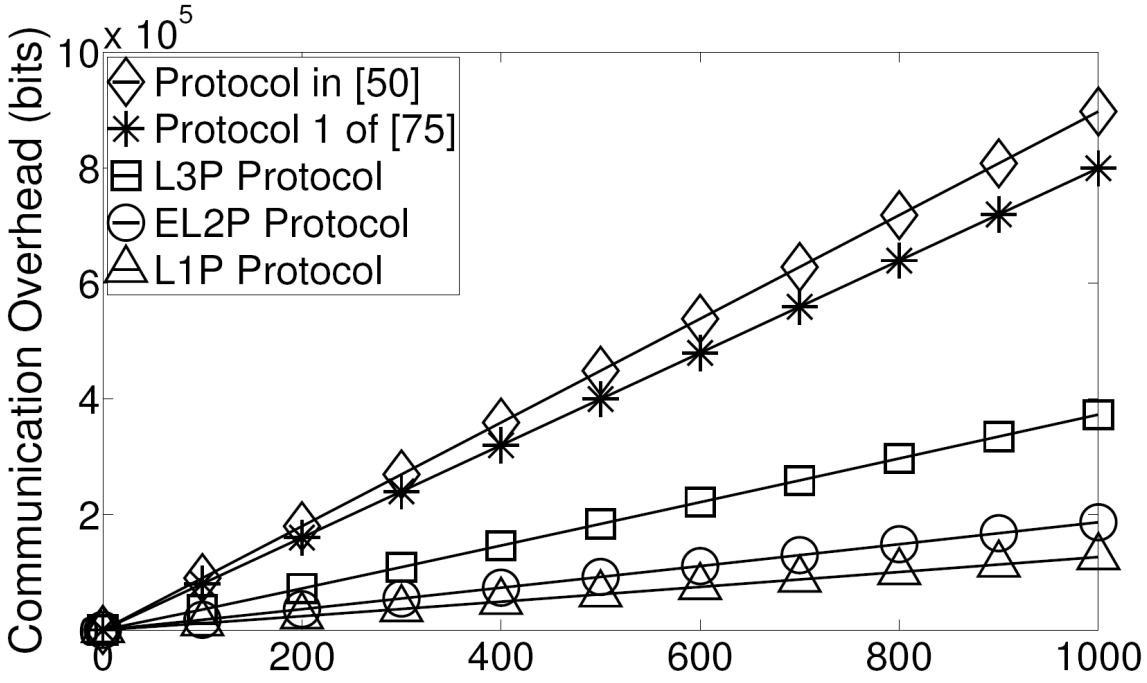

Figure 2.10

Comparison of total Computation Cost

2.5.2 Private Matching Protocols' Performance Evaluation

In this section, we evaluate the proposed protocols' performances in terms of computation cost, communication overhead, total running time, and energy cost, and compare them

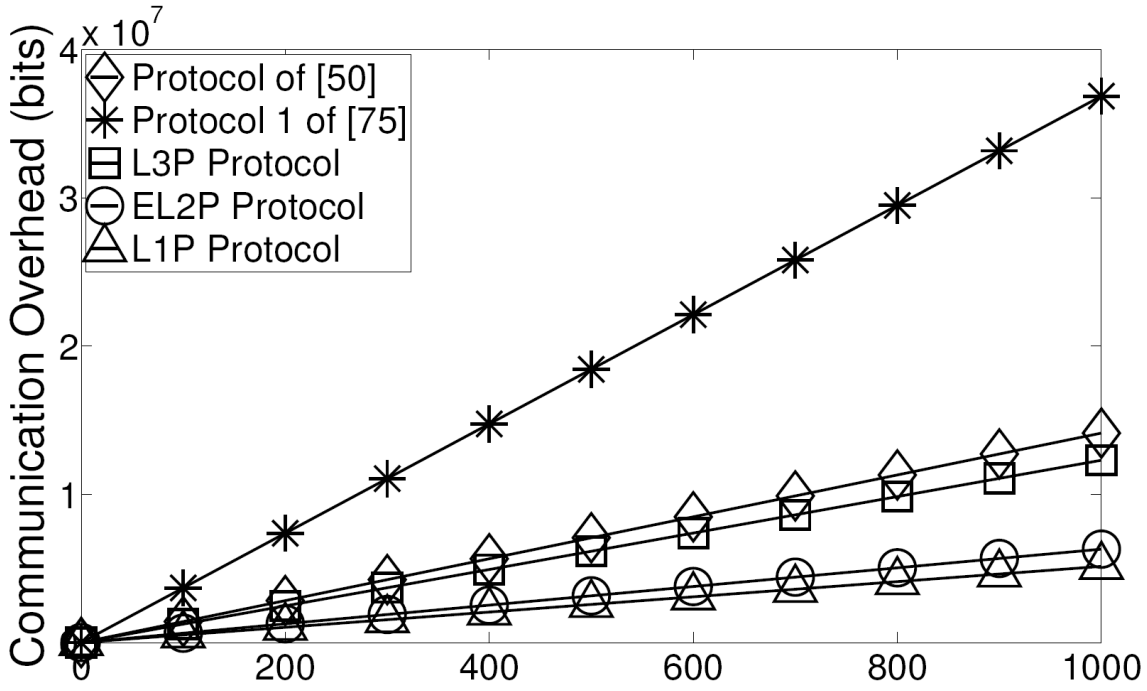

Figure 2.11

Comparison of total Communication Cost

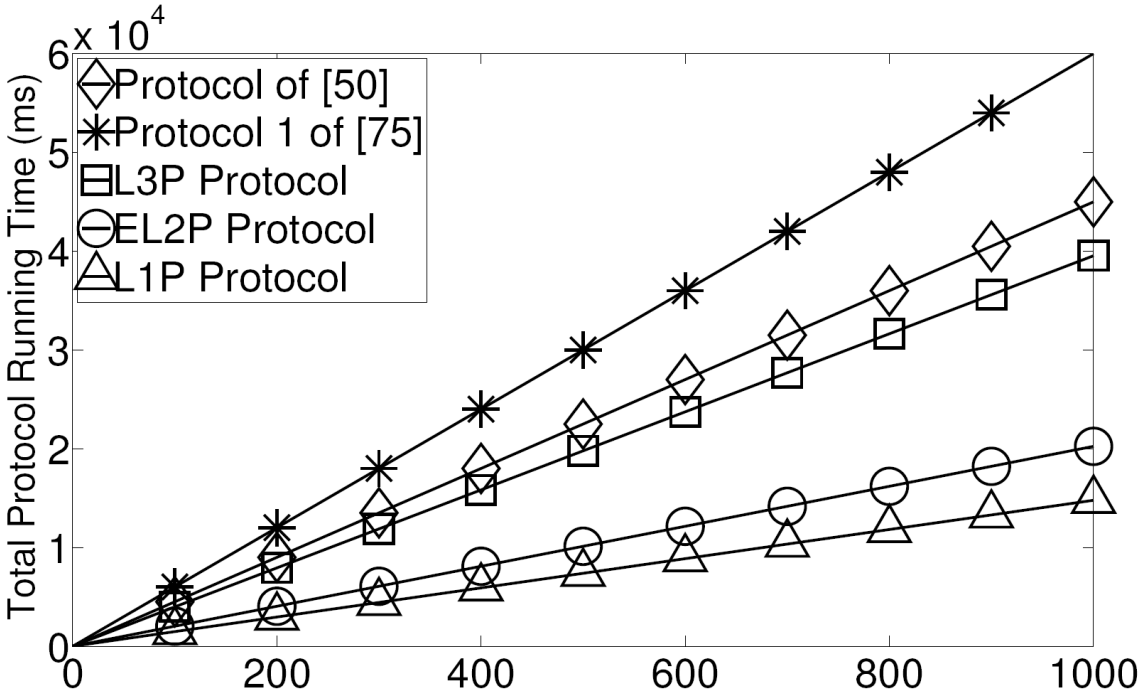

Figure 2.12

Comparison of total Protocol Execution Time

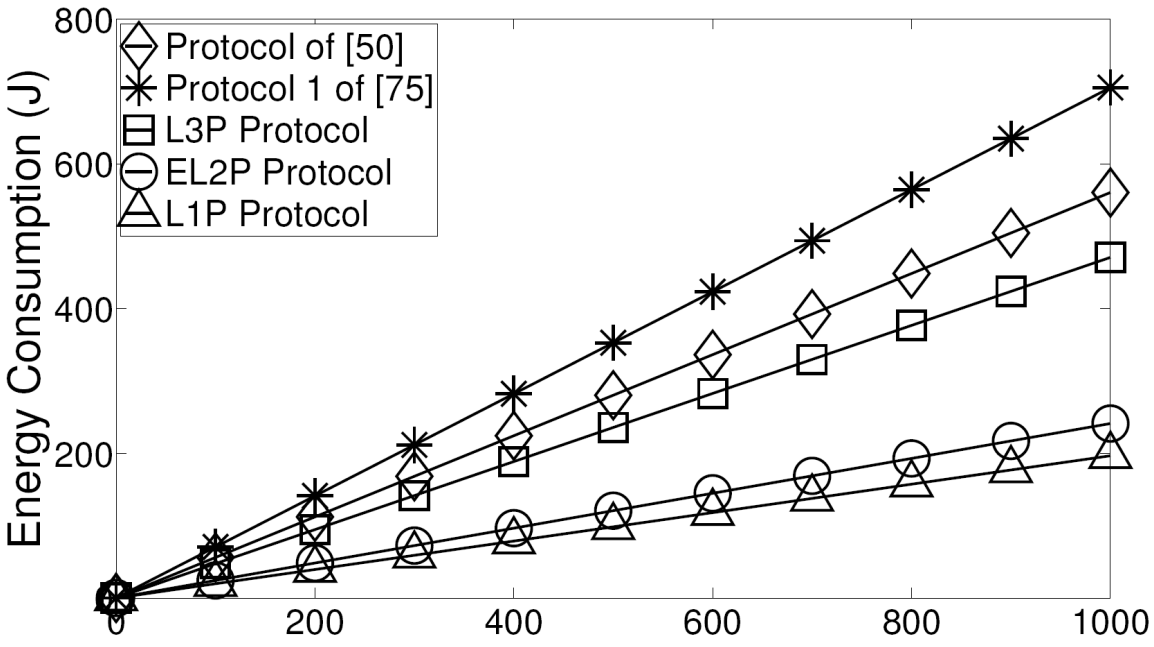

Figure 2.13

Comparison of total Energy Cost

with the performances of the protocols developed in [75] and [50]. In particular, Zhang et al. [75] present fine-grained private matching protocols using an additively separable function like l_1 norm. [75] defines d as the size of the public profile attribute set, which is the set of all possible profile attributes in an OSN. To conduct fair comparisons, we set the size of a user's overall community set equal to d in our proposed protocols. Another parameter γ in [75] denotes the range of the integer used to define a user's level of interest in a particular attribute in the public attribute set. For a reasonably fine-grade private matching, we consider $\gamma = 10$. Besides, [75] presents four protocols with comparable computation and communication complexity. We compare our protocols with their most efficient one: Protocol 1. Besides, Lin et. al [50] propose a privacy preserving friend searching protocol

where a user seeks to be introduced to another user's friends with certain attributes. We set attribute size $m = \gamma = 10$ and the number of friends equal to d for fair comparisons. The parameters for the elliptic curve cryptography in [50] are the same as those used in their paper, i.e. we use type D curve of the form $y^2 = x^3 + ax + b$ and the base field is represented by 160 bits.

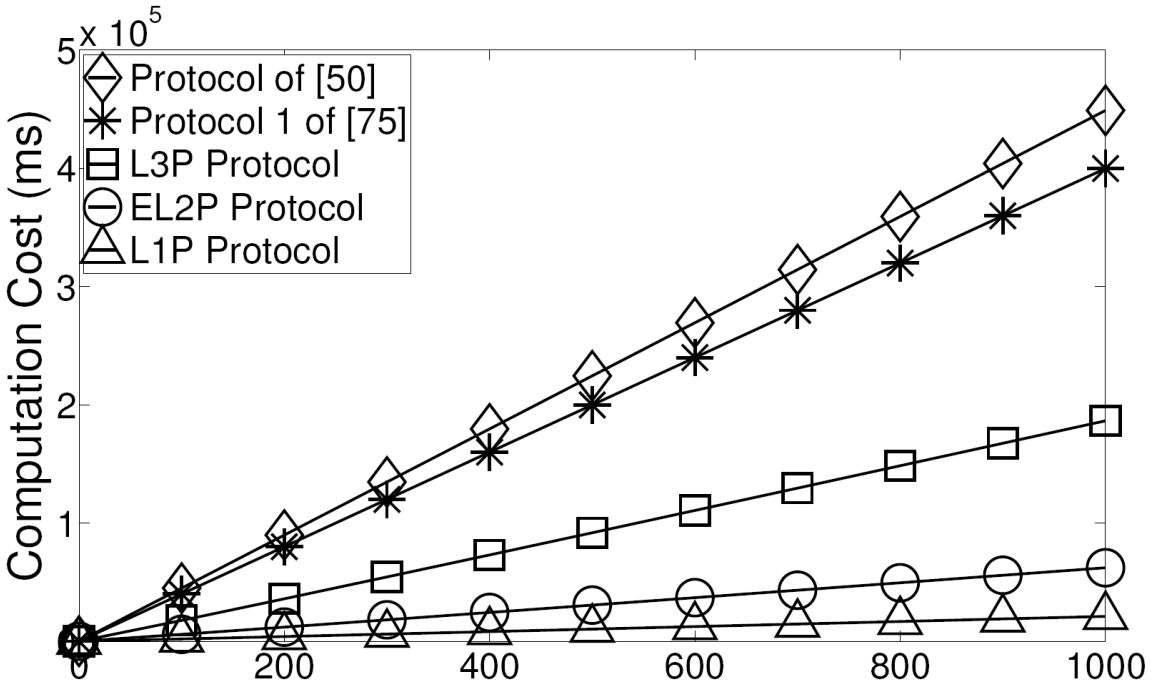

Figure 2.14

Comparison of the total Computation Cost of the Initiator

We have implemented our proposed protocols using a Java implementation of Paillier's cryptosystem [52]. We carry out simulations on a notebook with an Intel Core 2 Duo CPU and 2GB RAM. In the simulations, the same as that in [75], we focus on two wireless nodes communicating with each other, which both use IEEE 802.11 DCF as the MAC protocol

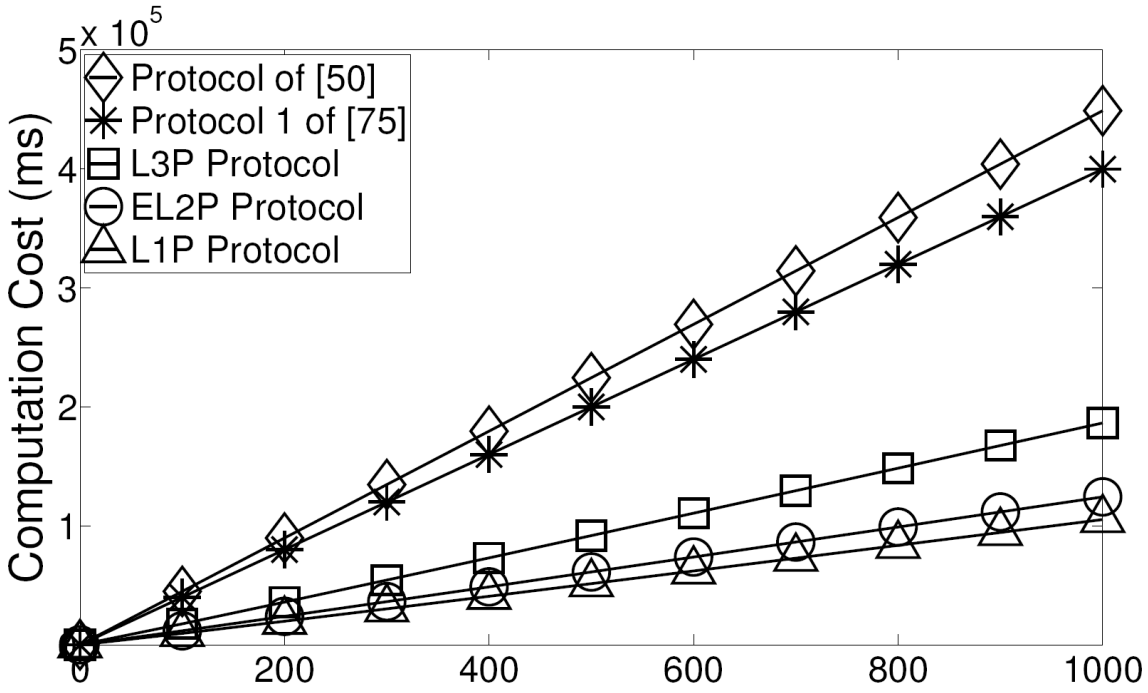

Figure 2.15

Comparison of the total Computation Cost of the Responder

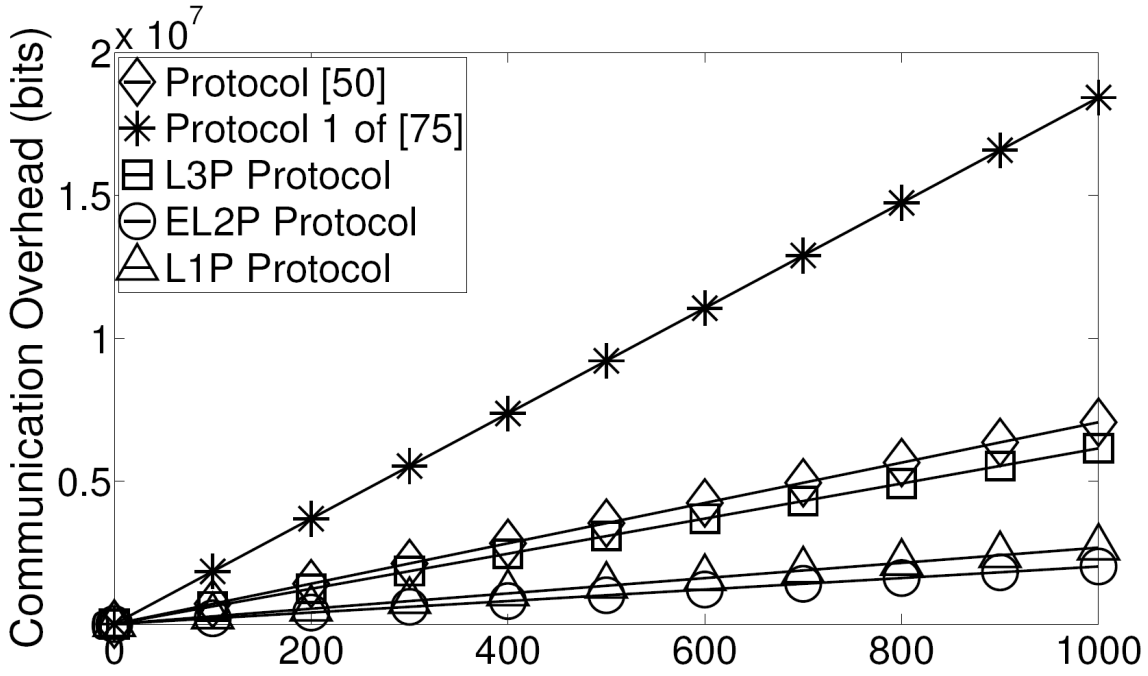

Figure 2.16

Comparison of the total Communication Cost of the Initiator

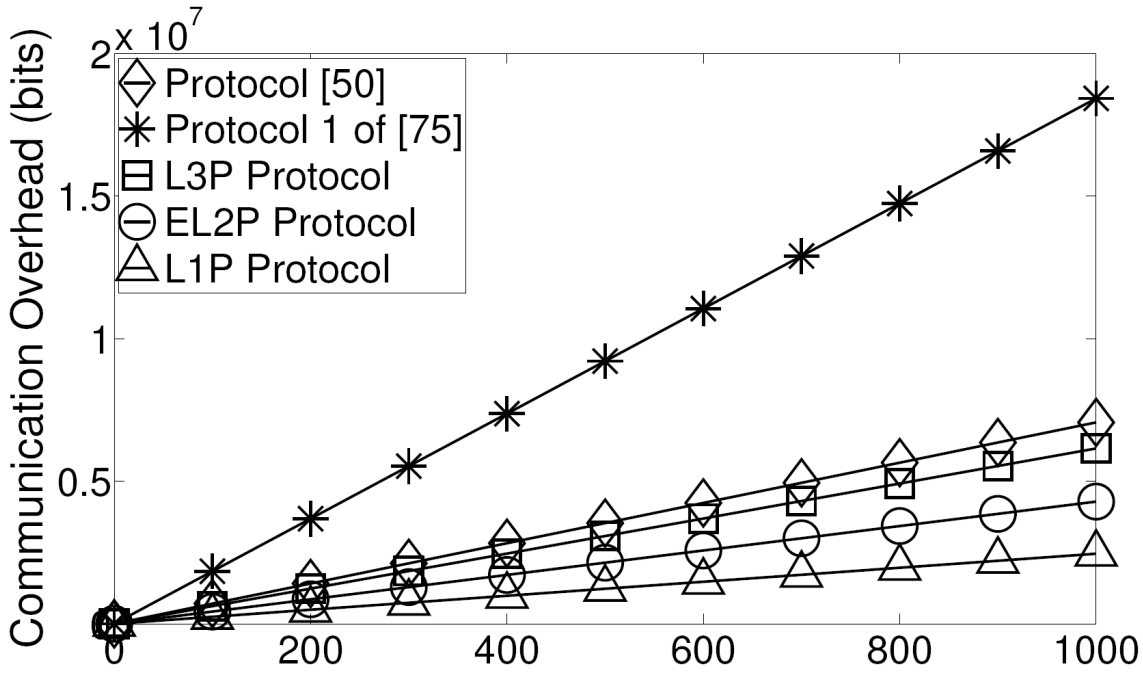

Figure 2.17

Comparison of the total Communication Cost of the Responder

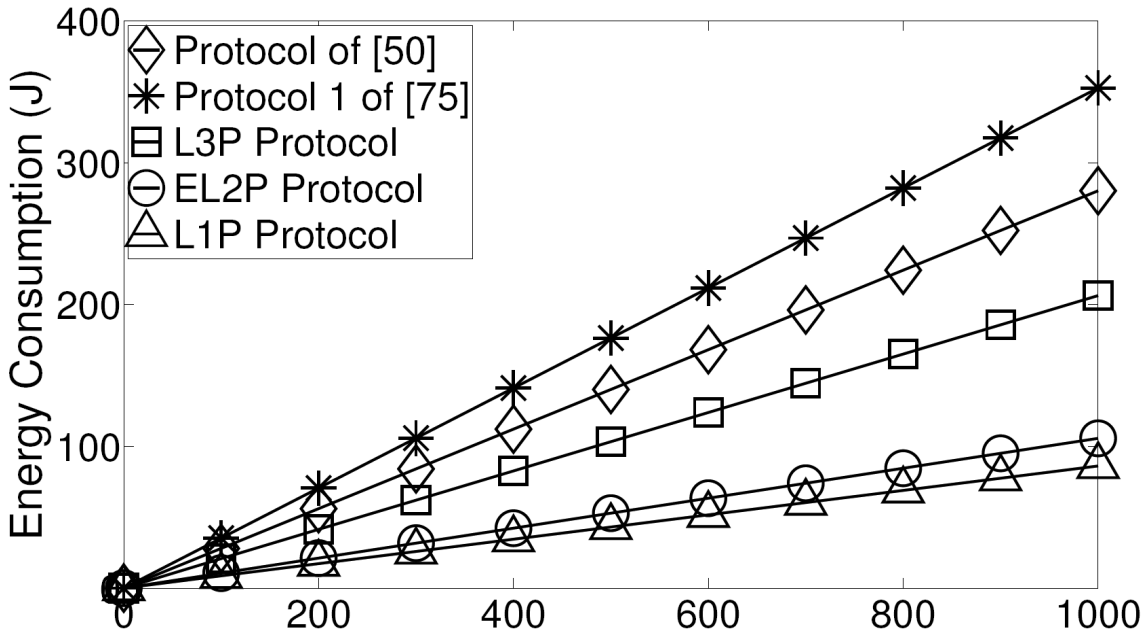

Figure 2.18

Comparison of the total Energy Cost of the Initiator

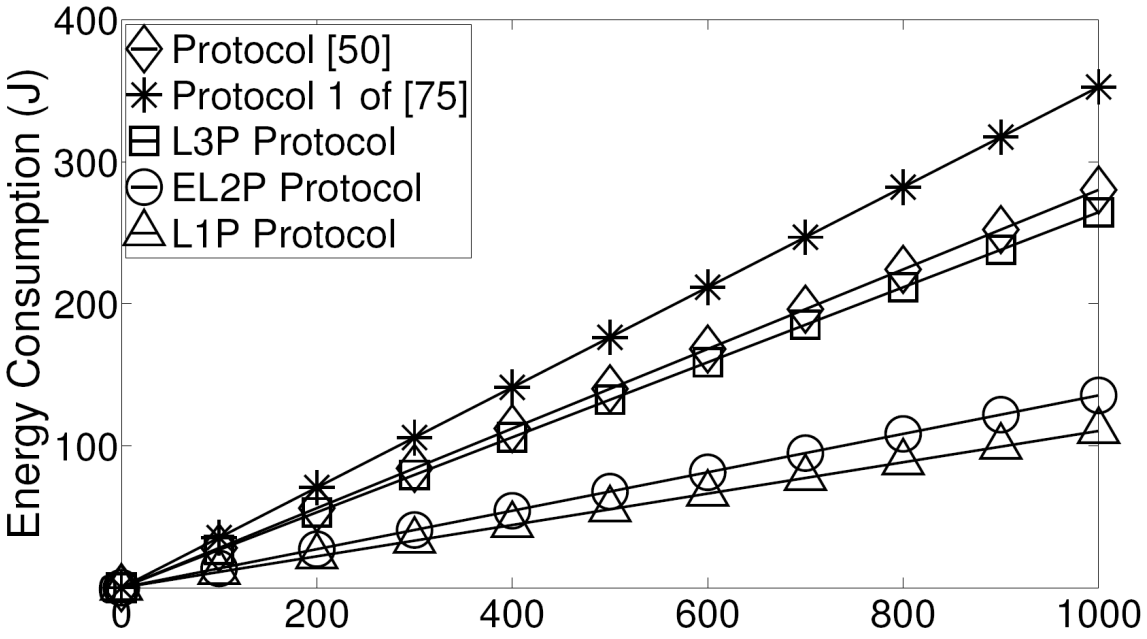

Figure 2.19

Comparison of the total Energy Cost of the Responder

with a data rate of 2Mbps. Besides, the energy consumption analysis neglects the energy consumed in computation and only considers the energy cost due to communications. In particular, we follow the energy model in QualNet [2] and assume the *Transmission*: *Reception*: *Idle* energy consumption ratios are 1.57 : 1.14: 1 [49].

We conduct two sets of simulations in this study. In the first simulation, we vary the size of the overall community set/the public profile attribute set/the number of friends while keeping the percentage of shared community constant at 10%, and γ, m at 10. For simplicity, we consider the Initiator and the Responder have the same overall community set size. Figure 2.5.1 compares the total of online and offline computation cost [5]. We can see that our most expensive protocol L3P has much lower computation cost than the most efficient protocol, Protocol 1, of [75] and the protocol of [50]. The reason is that each party in [75] needs to compute $O(d\gamma)$ exponentiations which is very expensive. Similarly, a larger number of ciphertexts due to Zero Knowledge Proof of Knowledge (ZKPoK) and blind key extraction in [50] result in a higher computation complexity. As shown in Figure 2.5.1, the communication overhead in [75] and [50] increases faster than our protocols when d/the number of friends increases. The communication cost of [50] is lower than that of [75] because of the smaller size of ciphertexts. Figure 2.5.1 compares the total protocol running time. Note that parallel processing between the communication and computation is implemented whenever possible in the protocols. Besides, the total running time takes into account the packet overheads at different layers. In addition, Figure 2.5.1 shows the

[5]The parameters for Figure 2.10 to Figure 2.19 are: $|\overline{C}_I| = |\overline{C}_R| = d$ = No. of friends, $|\overline{C}_I \cap \overline{C}_R| = 10\% \cdot |\overline{C}_I|(|\overline{C}_R|)$ and $\gamma = m = 10$. Similarly, the X-axis represents size of overall community set ($|\overline{C}_I| = |\overline{C}_R|$), size of public profile attribute set (d), and No. of friends in the proposed protocols, [75], and [50] respectively.

energy consumption of the protocols. We can easily find that our protocols require less running time and consume less energy than Protocol 1 in [75] and the protocol in [50]. We further extend the first experiment by breaking down the computation, communication, and energy costs to those for the Initiator and those for the Responder as shown in Figure 2.5.2-Figure 2.5.2. Note that the protocols in [75] and [50] need to run twice in order for both the Initiator and the Responder to obtain the private matching results, and the cost for the Initiator and that for the Responder are the same. We can see that both the Initiator and the Responder are subject to lower costs in our protocols.

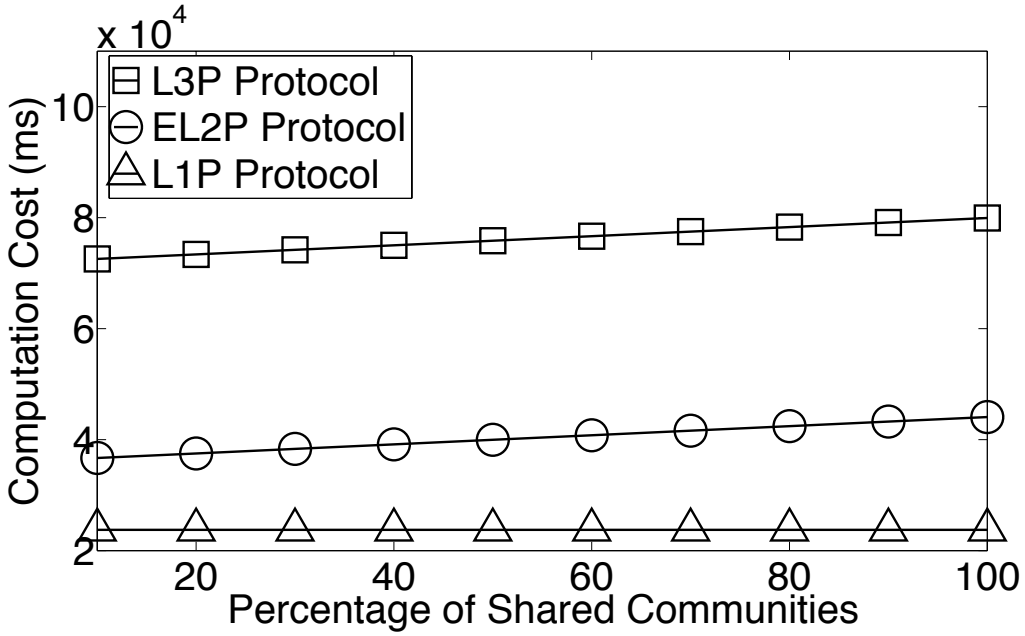

Figure 2.20

Performance comparison with varying size of the percentage of the shared communities$|\overline{C}_I \cap \overline{C}_R|$ ($|\overline{C}_I| = |\overline{C}_R| = 200$).

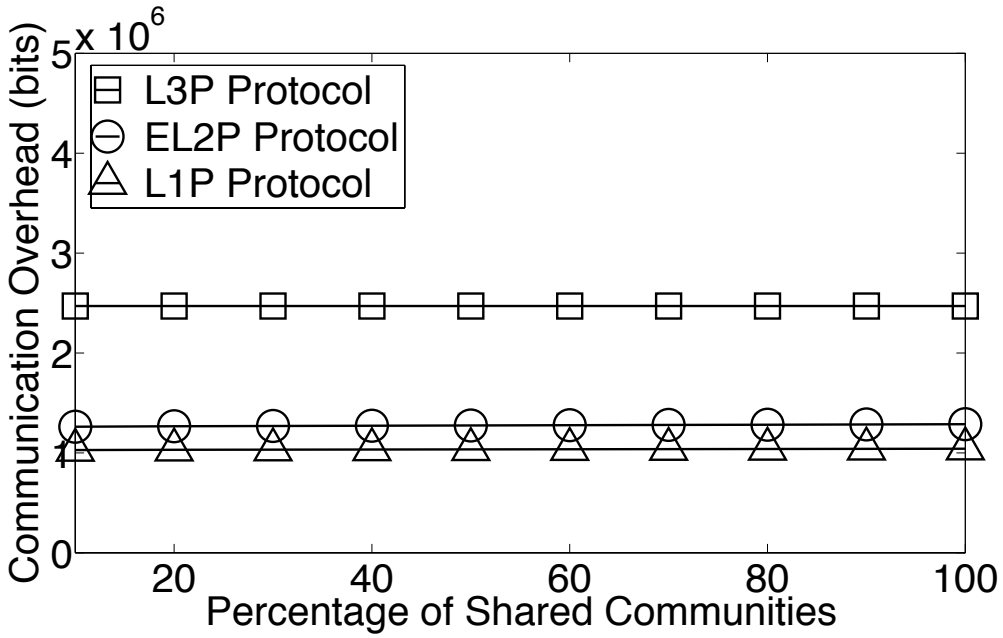

Figure 2.21

Performance comparison with varying size of the percentage of the shared communities $|\overline{C}_I \cap \overline{C}_R|$ ($|\overline{C}_I| = |\overline{C}_R| = 200$).

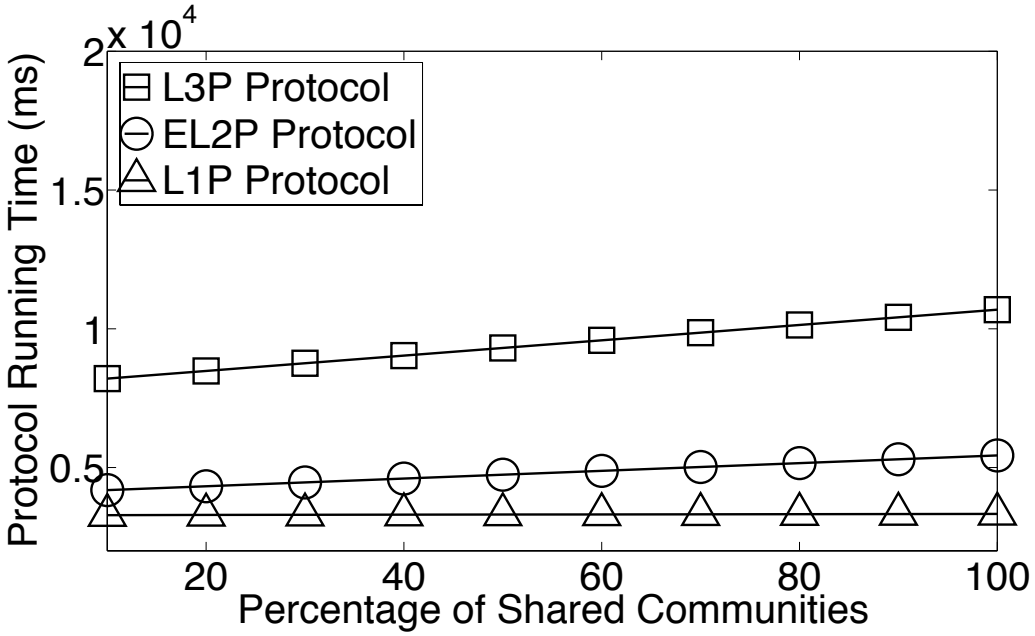

Figure 2.22

Performance comparison with varying size of the percentage of the shared communities $|\overline{C}_I \cap \overline{C}_R|$ ($|\overline{C}_I| = |\overline{C}_R| = 200$).

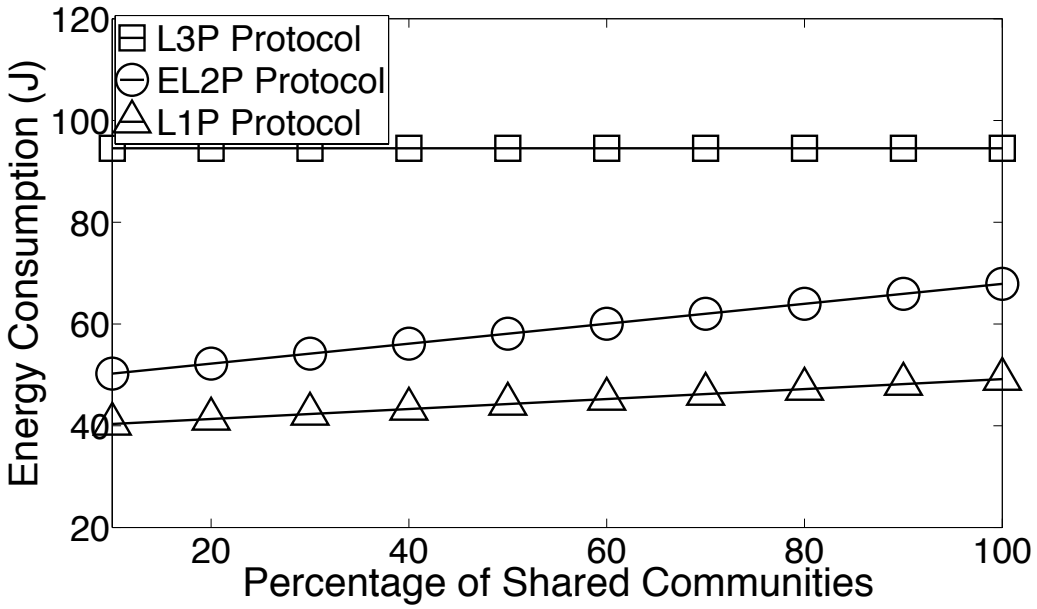

Figure 2.23

Performance comparison with varying size of the percentage of the shared communities $|\overline{C}_I \cap \overline{C}_R|$ ($|\overline{C}_I| = |\overline{C}_R| = 200$).

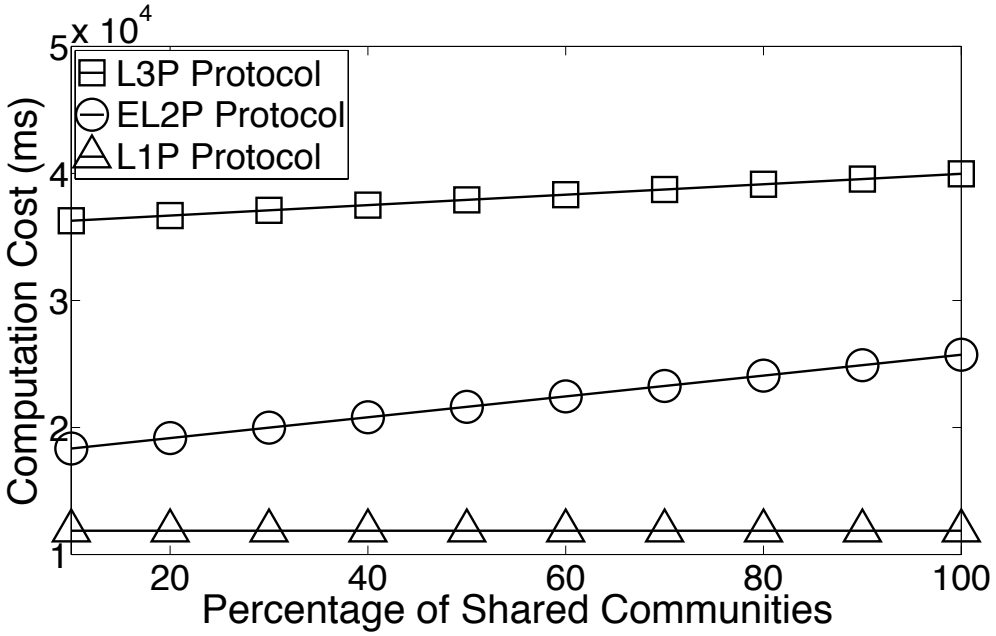

Figure 2.24

Computation cost for the Initiator with varying size of the percentage of the shared communities$|\overline{C}_I \cap \overline{C}_R|$ ($|\overline{C}_I| = |\overline{C}_R| = 200$).

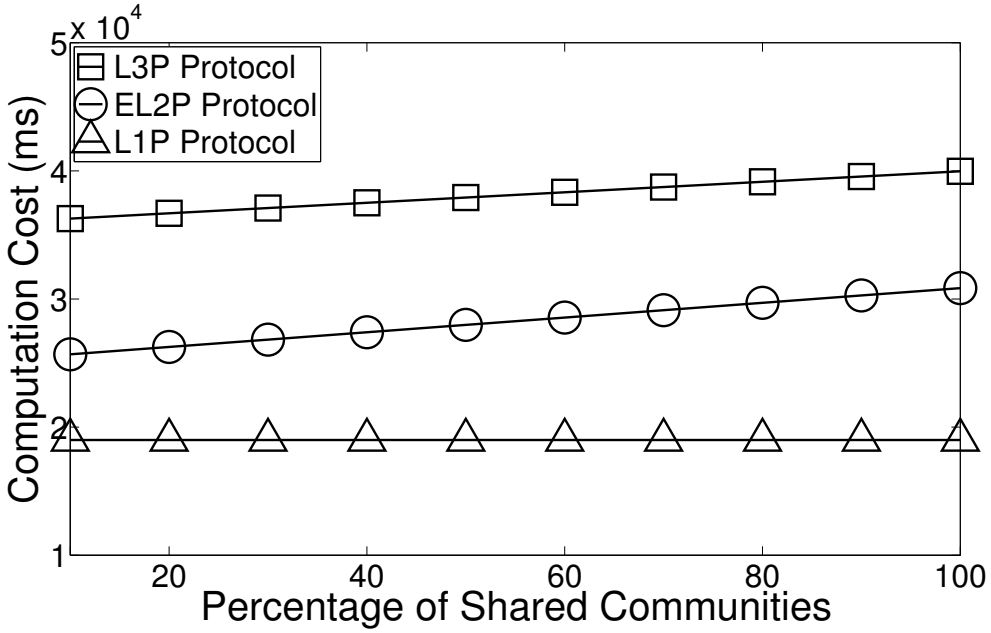

Figure 2.25

Computation cost for the Responder with varying size of the percentage of the shared communities $|\overline{C}_I \cap \overline{C}_R|$ ($|\overline{C}_I| = |\overline{C}_R| = 200$).

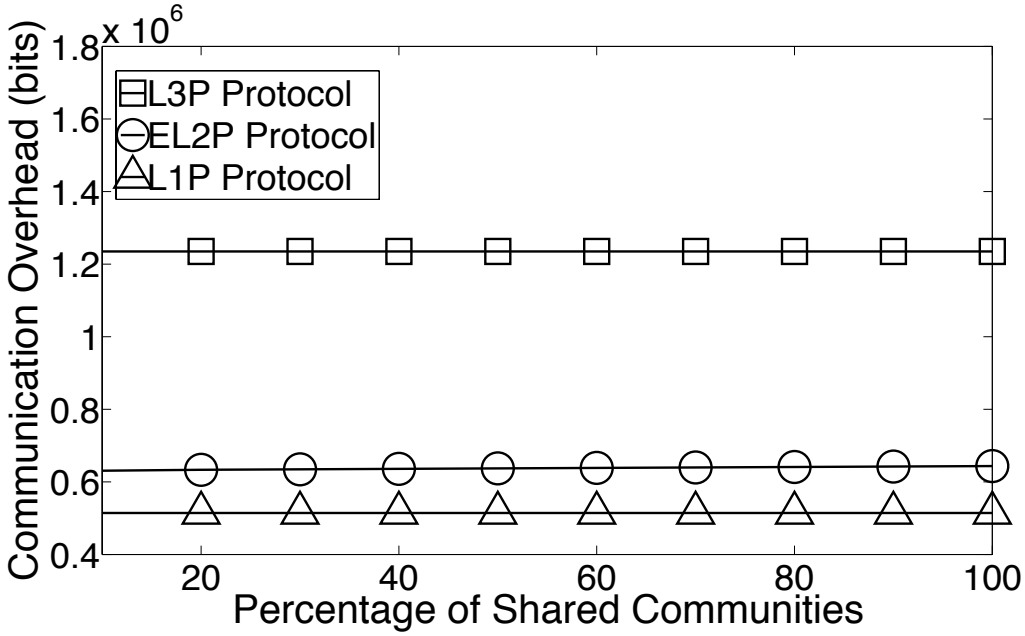

Figure 2.26

Communication cost for the Initiator with varying size of the percentage of the shared communities $|\overline{C}_I \cap \overline{C}_R|$ ($|\overline{C}_I| = |\overline{C}_R| = 200$).

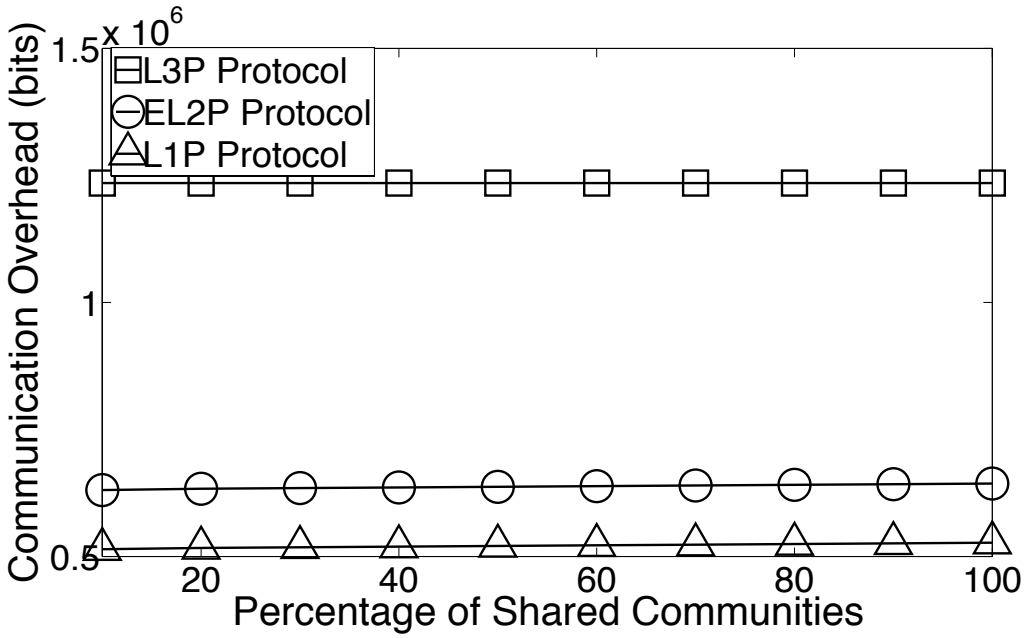

Figure 2.27

Communication cost for the Responder with varying size of the percentage of the shared communities$|\overline{C}_I \cap \overline{C}_R|$ ($|\overline{C}_I| = |\overline{C}_R| = 200$).

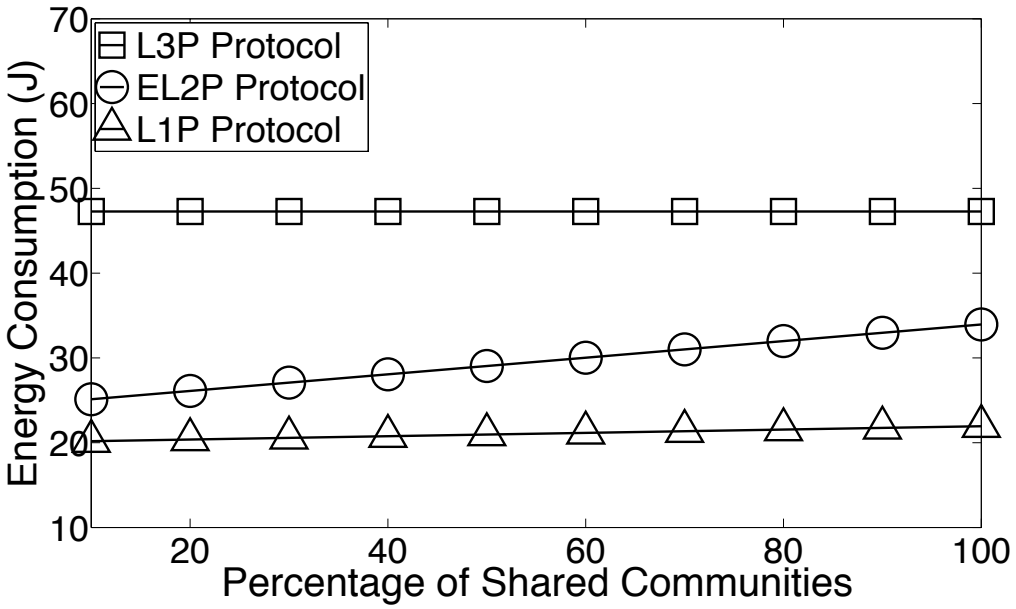

Figure 2.28

Energy cost for the Initiator with varying size of the percentage of the shared communities $|\overline{C}_I \cap \overline{C}_R|$ ($|\overline{C}_I| = |\overline{C}_R| = 200$).

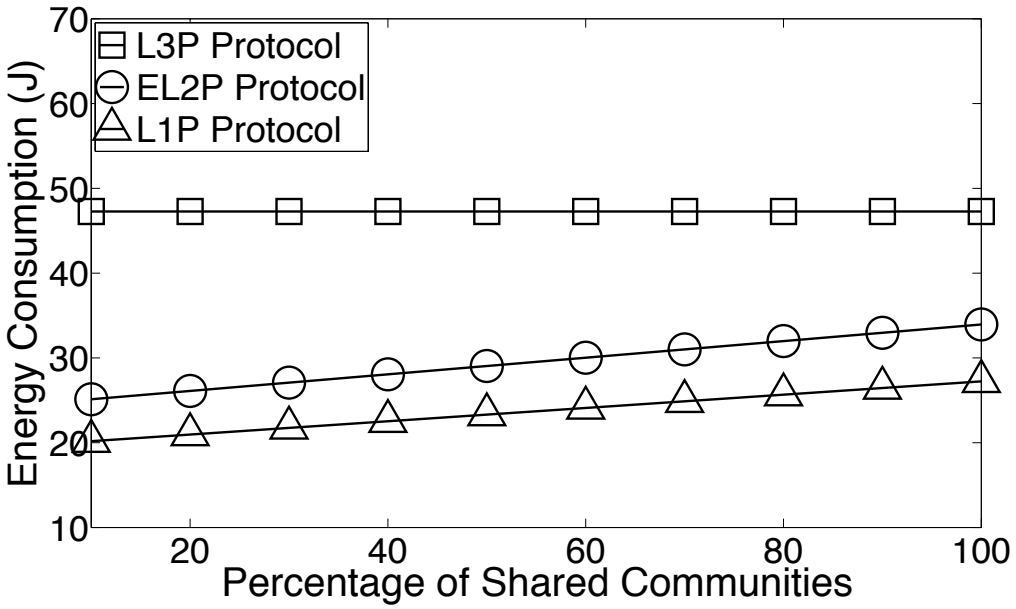

Figure 2.29

Energy cost for the Responder with varying size of the percentage of the shared communities $|\overline{C}_I \cap \overline{C}_R|$ ($|\overline{C}_I| = |\overline{C}_R| = 200$).

In the second set of simulations, we analyze the performance of our protocols when the percentage of common communities between the Initiator and the Responder varies between 0% and 100%. Figure 2.5.2 shows the computation cost. In particular, L1P's computation cost is irrelevant to the percentage of common communities since the computation cost of L1P does not depend on $|\overline{C}_I \cap \overline{C}_R|$. Besides, both EL2P's and L3P's computation cost increase only a little as $|\overline{C}_I \cap \overline{C}_R|$ increases. Regarding the communication overhead, Figure 2.5.2 shows that the communication overhead of the three protocols almost remains the same even when $|\overline{C}_I \cap \overline{C}_R|$ increases from 0% to 100% of the size of the overall community set. Fig. 2.5.2 shows the impact of the percentage of common communities on the total protocol running time. EL2P and L3P experience slight increase in total protocol running time since the computational (and communication too in L2P) overhead increases with the increase of $|\overline{C}_I \cap \overline{C}_R|$. Similarly, there is slight increase in the energy consumption of L1P and of EL2P when the fraction of the common communities over the size of the overall community set increases as shown in Figure 2.5.2. The energy consumption of L3P remains constant since there is no increase in communication, and hence no additional energy consumption, when the percentage of common communities increase. We further divide the computation, communication, and energy cost in this set of experiments into the corresponding cost incurred by the Initiator and the Responder in Figure 2.24-Figure 2.29.

2.6 Conclusion

The ever increasing use of OSNs has introduced a new paradigm in interacting with existing friends and making new friends in online world as well as in real life. Current schemes lead to privacy breaches. How to enable people to explore new friends in OSNs while preserving their privacy is an important and challenging problem. In this chapter, we have exploited the community structure of an OSN to define a realistic asymmetric social proximity measure, and presented three efficient protocols for privately computing the social proximity between two users in OSN. We have validated the proposed measure using real social network data and the simulation study shows the efficacy and the efficiency of the schemes compared to the state-of-the-art schemes.

CHAPTER 3

SPA: A SECURE AND PRIVATE AUCTION FRAMEWORK FOR DECENTRALIZED ONLINE SOCIAL NETWORKS

3.1 Introduction

E-commerce has exploded in the last decade. It enables the buying and selling of goods and services via electronic channels, primarily the Internet, and has become an indispensable part of our daily lives. On the other hand, the security and privacy threats on e-commerce are also on the rise. The threat model includes not only the usual malicious attacker(s), but also the advanced persistent threat (APT) and/or global surveillance capability from very resourceful entities/attackers. Particularly, attackers may compromise the e-commerce service providers' servers and steal consumers' confidential information like their personal data and buying/selling history. Such information can be used for many privacy intrusive purposes like directed marketing, user profiling. Besides, in auction based e-commerce like eBay, users' bidding statistics reveal their valuations for the items being auctioned and the server can utilize the statistical information to increase its financial gain in future auctions of similar items.

The explosive growth of online social networks (OSNs) over the past several years has dramatically changed the way information is produced and propagated in the world, and has made OSNs potential new great marketplaces for e-commerce. In particular, in OSNs,

the traditional unidirectional information flows, where information (e.g., breaking news, events) flows from a source (e.g., news organizations) to the consumers are being replaced with multidirectional flows, where the ordinary users of OSNs (like Facebook, Twitter, Youtube) are both the sources and the consumers of the information. This shift in information paradigm has been proven to be powerful in strengthening the connections among users, and hence can facilitate large-scale e-commerce. However, OSNs also raise serious concerns about users' privacy since the traditional OSNs store users' private personal, historical, and relationship information. For instance, Twitter was attacked in early January 2013 and about 250,000 user accounts might have been compromised, with names and e-mails possibly being uncovered [12]. Facebook, Apple, Microsoft were under similar attacks in February 2013 [37]. Thus, auctions in traditional OSNs will inevitably lead to security and privacy problems. Recently, decentralized online social networks (DOSNs) like [8, 18], and Dispora [1], attract users' intensive attention, where users own and store their private data on their own computers or the servers they trust. In this work, we exploit the rich connectivity and the distributed system architecture of DOSNs to develop a secure and private auction framework called SPA, which requires no trust among the participants for the privacy and correctness of auction outcomes.

The proposed auction framework SPA is based on Vickrey auction [68]. Specifically, auctions are frequently employed for determining resource allocations and selling prices. Many auction schemes have been proposed in the literature (please refer to [13] for a review on auction protocols). Vickrey auction adopted in this study, also known as second price auction, is a sealed-bid auction, in which bidders send their sealed bids to a trusted

auctioneer. The winning bidder of an auction is the highest bidder and is charged the second highest bidding price. Due to this pricing mechanism, it can be proved [68] that the bidders in Vickrey auction have the highest pay-off when they bid their true valuations of the goods being auctioned and hence have no incentives to bid strategically. Vickrey auction has attracted a lot of research interests because of this interesting property. However, it has rarely been used in practice mainly due to the following reasons. First, bidders hesitate to reveal their true valuations to the auctioneer who may exploit such information for privacy invasive purposes as mentioned before. Second, if the auctioneer is dishonest, it can cheat the winning bidder by creating an artificial second highest bid with bid price just lower than the highest bid since it knows all users' bidding prices.

The remedy to the problem of limited usage of Vickrey auction is to ensure each individual bidder that first, his/her bidding privacy (e.g., identities, bidding prices, histories) is preserved regardless of the number of possible other colluding bidders, and second, there is no need to bank on the honesty of the auctioneer for the correctness of auction outcomes. Our auction framework SPA consists of three phases: identity initiation, buyer-seller matching, and private auction. It can guarantee users' privacy and auction correctness, while only revealing minimum information, i.e., the winning price and the winning bidder's public pseudo identity.

Specifically, in the identity initiation phase, each user who would like to participate in an auction obtains a public and a private pseudo identity (ID) from a Trusted Third Party. In the "buyer-seller matching" phase, we develop an efficient algorithm to enable the users interested in buying/selling item(s) to distribute their intents and match each other. In par-

ticular, the users utilize both social links and the underlying Distributed Hash Table (DHT) links to route their advertisement messages to a randomly chosen user, called *the bridge node*, who then helps match the buyers and sellers. In the private auction phase, we design an efficient bidder-resolved private auction protocol. Particularly, bidders use their public pseudo IDs to get authenticated through non-interactive zero knowledge (NIZK) proofs. Thus, their ID privacy can be protected. Then, the authenticated bidders collaboratively construct a public encryption key based on a distributed exponential Elgamal cryptosystem. They send their encrypted bidding vectors to the bridge node, which can thus be protected against attacks like colluding. The bidders also sign their encrypted bids with an anonymous signature scheme so that the bids are non-repudiable. After that, the bridge node calculates the winning price under public scrutiny, without revealing any bidder's bidding vector. The winning bidder can finally be determined without revealing his/her bidding vector.

Besides, we analyze the computation and communication complexities of the proposed private auction scheme, which are $O(n + K)$ for each node where n is the number of bidders and K is the number of pricing points, while those of previous auction schemes (without the winning bidder identification process) [7] are $O(nK)$ at best. Security and privacy of SPA are also investigated.

We summarize our major contributions in this work as follows.

- To the best of our knowledge, the proposed auction framework SPA is the first attempt to address auctions in DOSNs.
- We design a distributed private buyer-seller matching scheme to enable auctions in DOSNs where no central server or auctioneer is available. The communication cost is $O(\log n)$ where n is the network size.

- We develop a fully private distributed auction protocol, whose computation and communication complexities are both $O(n+K)$ for each node where K is the dimension of a seller's price vector. In contrast, the most efficient existing distributed private auctions like [7] are not fully private and have higher complexities of $O(nK)$.
- SPA can provide security, privacy, authenticity, non-repudiation, and correctness for the auctions.
- We implement our auction protocol and show that it outperforms previous auction schemes significantly in terms of both computation and communication costs.

The rest of the Chapter is organized as follows. In the next section we briefly discuss the existing works that are most related to our work. Section 3.3 presents the system model, adversary model, and design goals. Section 3.4 introduces the cryptographic and DHT preliminaries for our protocol design. Section 3.5 describe the detailed design of the proposed SPA auction framework. We analyze the computation/communication complexity and privacy/security in Section 3.6. We present the performance evaluation results in Section 3.7 and finally conclude the Chapter in Section 3.8.

3.2 Related Work

In this section, we introduce the important existing works that are most relevant to our study.

3.2.1 Decentralized Online Social Networks (DOSNs)

Currently most OSN service providers like Twitter, Facebook, Google$^+$ use central servers to store users' private data, which, however, raises great concerns about users' privacy. For example, Twitter was attacked in early January 2013 and about 250,000 user accounts might have been compromised, with names and e-mails possibly being uncovered [12]. Facebook, Apple, Microsoft were under similar attacks in February 2013 [37].

Besides, users may not have connectivity to the server all the time. Thus, recently the research on decentralized online social networks (DOSNs) has attracted intense attention. There have been several proposals for DOSNs [8, 18] in the literature and some (e.g., [1]) have taken off and are increasingly popular. Specifically, PeerSoN [8] has a two-layer architecture where peers (with social relationships) communicate with each other using a distributed harsh table (DHT) based lookup service. Safebook [18] proposes to build concentric rings of nodes around each node, based on the degree of trust among nodes, to provide trusted data storage, profile data retrieval, and communication obfuscation through indirection. Diaspora [1] is a popular DOSN, where users can host their data on their own computers or in the servers they trust.

3.2.2 Distributed Hash Table

Distributed Hash Tables (DHTs) have been used as efficient lookup schemes in peer-to-peer systems. In particular, each peer is assigned a unique ID and keeps a record of a small fraction (usually $\log n$, n is the network size) of the nodes in the network, which are determined by certain specific algorithm to guarantee efficient lookup service. For instance, Chord [66] uses consistent hashing [40] to assign node ID and to map a given key or data to a specific node. Other notable and widely referred DHT schemes include Kamedia [54], CAN [61], Pastry [62], and Tapestry [76]. Recently, DHT systems are designed while addressing security (especially against sybil attack) [45, 73] and anonymity [69, 70] issues. Our proposed private auction protocol utilizes DHTs (based on the Chord [66] protocol) and social links for efficient advertisement distribution and buyer seller matching.

3.2.3 Cryptographic Auction Protocols

The necessity of providing security and privacy for the participants of an auction has led to intensive research activities on cryptographic auction protocols. Yao's garbled circuit [72] based multi-party computation (MPC) is tailored to design secure auction [39,58] with multiple (two) auctioneers under a passive adversary model, where the two auctioneers are assumed not to collude with each other. Similar threshold based MPC auction protocols [33, 42, 63] rely on multiple auctioneers and are secure as long as there are no more than a certain fraction of the total number of auctioneers colluding with each other. Lipmaa et al. [51] employ homomorphic encryption to design a secure Vickrey auction scheme, where the semi-honest auctioneer therein knows all users' bidding prices.

A few approaches have been developed to improve the privacy in auction. Brandt [6] propose a private auction scheme, in which the bidders engage in cryptographic protocols and jointly compute the outcome of an auction, and later improve the protocol in [7]. In such auctions, collusion between any numbers of bidders but the total number of bidders is insufficient to compromise the auction privacy. The computation complexity and communication complexity of the Vickrey based auction scheme in [7] are both $O(nK)$, where n is the network size and K is the number of possible bidding values. However, Dreier et al. [23] show that the bid privacy in [7] can be breached if interactive Zero Knowledge Proof (ZKPs) are used. More importantly, even if Non-Interactive ZKPs (NIZKPs) are used, due to the lack of authentication, malicious bidders can mount a collaborative attack to breach the privacy of a targeted bidder. In contrast, our proposed auction protocol is a fully private auction protocol with both communication and computation complexi-

ties being $O(K)$ ($O(K)$ if with the winner identification process), and hence much more efficient.

3.3 Problem Formulation

3.3.1 System Model

We consider a Decentralized Online Social Network (DOSN) consisting of three layers as shown in Figure 3.1. The OSN layer at the top includes social network users along with the relationships among them. Particularly, an OSN can be defined as a graph $G = (V, E)$, where the set of vertices $V = \{v_1, v_2, ..., v_n\}$ represent nodes (users) in the network and the set of undirected edges $E = \{e_{ij}\}$ ($1 \leq i, j \leq n$, $i \neq j$) represent the friendships or social ties among the nodes. In the absence of a central server, the Distributed Hash Table (DHT) layer provides the peer-to-peer lookup functionality in the DOSN, which we utilize to distribute the advertisement messages of buyers and sellers. Unlike OSN links, the DHT links are directed. We build the DHT links based on Chord DHT protocol [66], which will be briefly introduced in Section 3.4.3. Each node i has a "Chord ID" at the DHT layer denoted by u_i. The actual communications take place at the Internet layer at the bottom. Each user in the DOSN is a potential buyer/seller and has a public page where, if the user is selected as a bridge node (see Section 3.5), the information on the item for sale, the encrypted bids from the buyers, and auction related computations are hosted. A fixed time period (e.g., a day or a week) is determined for each auction during which buyers need to submit their bids to the bridge node.

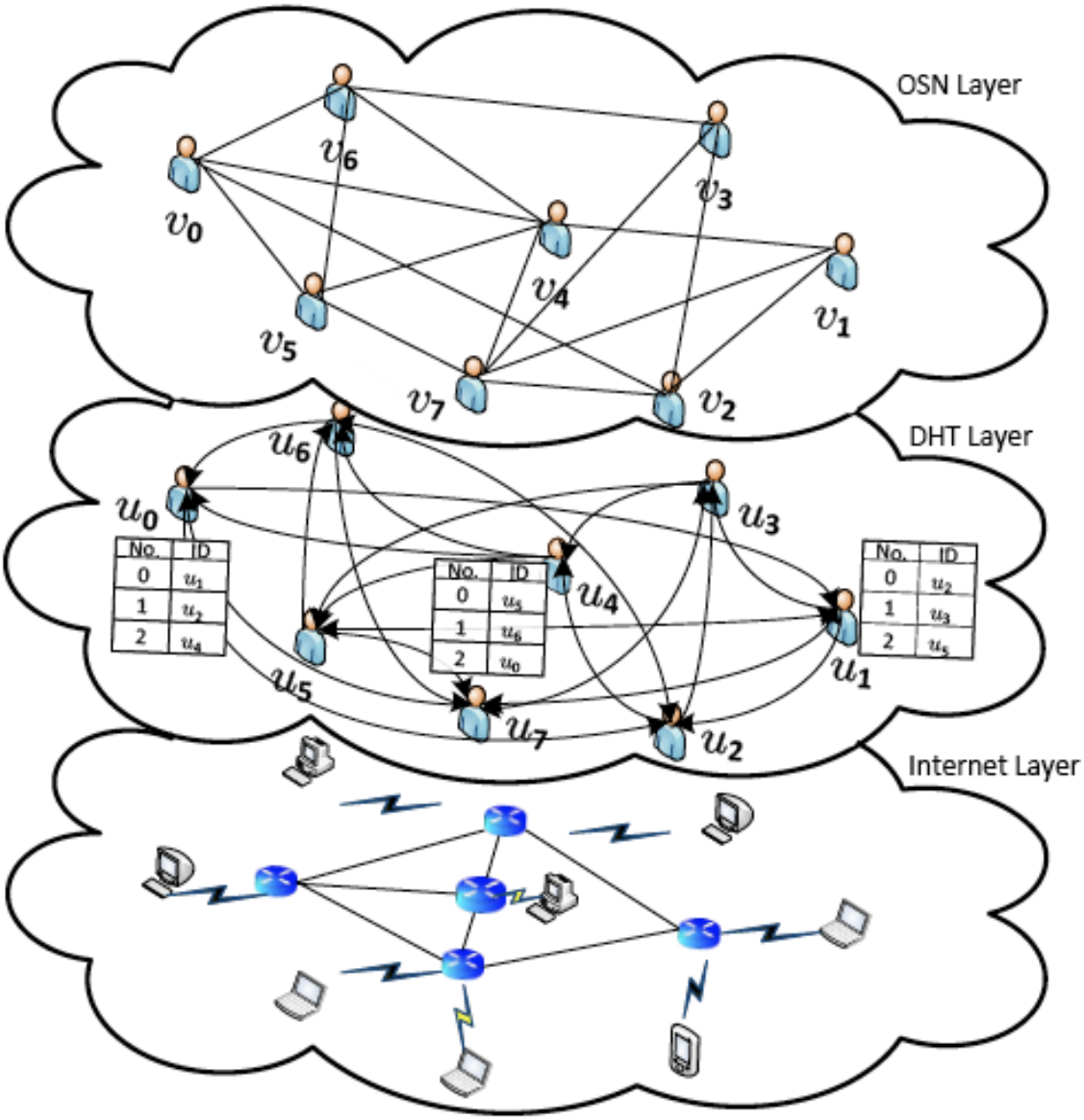

Figure 3.1

A Decentralized Online Social Network (DOSN) consisting of three layers.

3.3.2 Adversary Model

The adversaries taken into consideration in this work are mainly the participants in the auctions, such as bidders, sellers, auctioneers (i.e., bridge nodes). These participants may be interested in bidders' bidding prices in order to enhance their financial gain in the current auction or in the future auctions. For example, if a seller knows the bidding statistics of an item, he/she can exploit that information in future auctions to maximize his/her own financial gain. Similarly, bidders' identities can also be of interest to the adversaries, e.g., for targeted advertisement. Besides, the adversaries include the malicious bidders who may send bogus bidding values just to hinder the outcome of the auction. Note that we do not consider the possible adversaries at the DHT layer and Internet layer who may try to disrupt the auction by replaying or dropping the auction messages. There have been several works addressing such attacks [45, 73] and protecting privacy at the DHT layer [69, 70]. While our scheme can be easily built on these DHT protocols to provide security and privacy at the DHT and Internet layers, in this work we employ a widely referred DHT protocol [66] and mainly focus on the possible adversaries in our auction scheme as mentioned above.

3.3.3 Design Goals

Our design goals are summarized as follows:

- **Security**: The proposed system may be under various attacks such as impersonation, colluding. Our goal is to protect malicious attackers from disrupting auction outcomes by indulging in the intermediate computations in the protocol.
- **Privacy**: In the proposed Vickrey based auction scheme, where buyers bid with their true valuations of the items being auctioned, the bid privacy is important to the buyers. All buyers' and sellers' identities should be protected too. Our goal is to provide privacy (e.g., bidding prices, identities) for users during and after the auctions and make sure that users' buying/selling histories cannot be tracked.

- **Authenticity and Non-Repudiation**: Since bidders' identities are hidden during the auctions to protect their privacy, we need to validate that the bidders are legal users in the system. We also need to verify that the bidding values are legit as they are unknown to the auctioneer. Besides, we aim to achieve non-repudiation in the auctions, i.e., guarantee that bidders cannot deny their bidding. Thus, our goal is to ensure authenticity and non-repudiation in the auctions.
- **Efficiency**: Social networks usually host a large number of users, all of whom can engage in auctions. Therefore, the communication/computation complexity of auction schemes should not increase rapidly with the number of participating users. Our goal is to obtain high efficiency in the auctions in terms of communication and computation complexities.

3.4 Preliminaries

3.4.1 ElGamal Cryptosystem

ElGamal cryptosystem [24] is a semantically secure homomorphic cryptosystem based on the intractability of the discrete logarithm problem in finite fields. In particular, let p and q be two large strong prime numbers such that $p = 2q+1$. Let $\mathbb{G}_q$ denote a sufficiently large multiplicative subgroup of $\mathbb{Z}_p^*$ with order q. A user chooses a random $x \in \mathbb{G}_q$ as the private key, and $y = g^x \mod p$ as the public key where g is a common generator of $\mathbb{G}_q$. All the calculations are modulo-p unless mentioned otherwise. A message $m \in \mathbb{G}_q$ for the user is encrypted as $Enc(m) = \langle \alpha, \beta \rangle = \langle g^r, my^r \rangle$, where $r \in \mathbb{G}_q$ is a local random number generated by the encrypting party. The user can then decrypt the message by calculating $Dec(\alpha, \beta) = \frac{\beta}{\alpha^x} = \frac{my^r}{(g^r)^x} = m$. ElGamal cryptosystem is multiplicative homomorphic, i.e., $Dec(Enc(m_1) \cdot Enc(m_2)) = Dec(\langle g^{r_1} \cdot g^{r_2}, m_1 y^{r_1} \cdot m_2 y^{r_2} \rangle) = m_1 \cdot m_2$. Additive homomorphism can be obtained with what is sometimes called "exponential" ElGamal, in which encryption is performed as $Enc(m) = \langle \alpha, \beta \rangle = \langle g^r, g^m y^r \rangle$ and decryption can be obtained by $Dec(\alpha, \beta) = \frac{\beta}{\alpha^x} = g^m$. Thus, $Dec(Enc(m_1) \cdot Enc(m_2)) = Dec(\langle g^{r_1} \cdot$

$g^{r_2}, g^{m_1} y^{r_1} \cdot g^{m_2} y^{r_2}\rangle) = g^{m_1+m_2}$. Note that since the decryption results in g^m instead of m, it is computationally intractable to obtain m from g^m due to the intractability of the discrete logarithm problem. The proposed auction scheme employs exponential ElGamal to utilize the additive homomorphic property, and only needs to determine whether m is zero which can be easily done.

Moreover, ElGamal cryptosystem can be used in distributed sysetms where there is strong privacy requirement. In particular, users can encrypt and decrypt messages in a distributed manner. A user can protect his/her privacy against the collusion among any number of users less than the total number of users. In the following, we describe how distributed encryption and decryption of ElGamal cryptosystem can be carried out.

- **Distributed Key Generation:** Each user v_i participating in the distributed key generation selects x_i as his/her private key and publishes $y_i = g^{x_i}$ as his/her public key. The public key for distributed encryption is then $y = \prod_{i=1}^{n} y_i = g^{\sum_{i=1}^{n} x_i}$.
- **Distributed Encryption:** A user can use the public key y to encrypt a message m. The resulting ciphertext is $\langle \alpha, \beta \rangle = \langle g^r, g^m y^r \rangle$.
- **Distributed Decryption:** All the users who participated in public key generation need to cooperate to decrypt the encrypted message. Specifically, if $\langle \alpha, \beta \rangle = \langle g^r, g^m y^r \rangle$ is the encrypted message, then each user publishes $\alpha_i = \alpha^{x_i}$. The original message can be recovered by any user by computing $\frac{\beta}{\prod_{i=1}^{n} \alpha_i} = \frac{g^m y^r}{g^{(\sum_{i=1}^{n} x_i) r}} = g^m$.

3.4.2 Zero Knowledge Proofs

The Zero Knowledge Proof (ZKP), introduced by Goldwasser, Micali and Rackoff (GMR) [31], is an important tool in cryptography. A prover can use a ZKP protocol to prove the possession of certain information to a verifier without revealing the very information. The absence of a trusted central authority in a DOSN makes the network inherently vulnerable to malicious users who aim to fulfill their malicious intents and do not follow

the proposed auction protocol. Besides, the strong privacy requirement in our schemes necessitates preserving bidders' anonymity and their bidding price privacy, which further complicates the authenticity and enforcement of correct protocol execution by all the participants. In order to ensure the bidders follow the proposed auction protocol correctly, we require all bidders (provers) to prove to a bridge node (verifier, see Section 3.5 for details) using ZKPs in different steps of the protocol. We describe several ZKPs we will use in SPA as follows. All the calculations are modulo-p unless mentioned otherwise.

3.4.2.1 Proof of Knowledge of A Discrete Logarithm

Schnorr [65] develops a ZKP that a prover (a bidder) can use to prove the knowledge of x such that $y = g^x$ to a verifier (a bridge node) who knows y and g.

- The bidder chooses a random r and sends $z = g^r$ to the bridge node.
- The bridge node sends a random challenge c to the bidder.
- The bidder computes $a = (r + cx)$ mod q and sends to the bridge node.
- The bridge node checks to see if $g^a = zy^c$.

If the equality holds, the bidder is able to prove to the bridge node the knowledge of x such that $y = g^x$ without disclosing x.

3.4.2.2 Proof of Equality of Two Discrete Logarithms

When a prover (a bidder) needs to prove that two values (encryptions, say $y_1 = g_1^x$ and $y_2 = g_2^x$) are computed using the same private key (x) to a verifier (a bridge node who knows y_1, y_2, g_1, g_2), the protocol below [11] can be employed to realize the zero-knowledge proof.

- The bidder chooses a random r and sends $z_1 = g_1^r$ and $z_2 = g_2^r$ to the bridge node.
- The bridge node sends a random challege c to the bidder node.
- The bidder then computes $a = (r + cx)$ mod q and sends to the bridge node.
- The bridge node checks to see if $g_1^a = z_1 y_1^c$ and $g_2^a = z_2 y_2^c$.

If both the equalities hold, the bridge node is convinced that the same x is used to compute y_1 and y_2.

3.4.2.3 Proof That An Encrypted Value Decrypts to Either 1 Or 0

In our private auction scheme (Section 3.5.3), a bidder prepares a bidding vector by encrypting each element (either 0 or 1) separately. While the actual bidding price (and bidding vector) remains private to the bidder throughout the auction, it is necessary to make sure the bidding vector is prepared correctly in order to deter any malicious bidder's attempt to disrupt the protocol. A bidder can use the protocol proposed by Cramer et al. [14] to prove to the bridge node that his/her bidding vector is composed of encryptions of $m \in \{0, 1\}$. Specifically, let $\langle \alpha, \beta \rangle = \langle g^r, g^m y^r \rangle$ be the ElGamal encryption of message m.

- If $m = 0$, the bidder chooses r_1, d_1, w at random and sends $\langle \alpha, \beta \rangle, a_1 = g^{r_1} \beta^{d_1}, b_1 = y^{r_1} (\alpha/g)^{d_1}$ and $a_2 = g^w, b_2 = y^w$ to the bridge node.
  If $m = 1$, the bidder chooses r_2, d_2, w at random and sends $\langle \alpha, \beta \rangle, a_1 = g^w, b_1 = y^w, a_2 = g^{r_2} \beta^{d_2}$, and $b_2 = y^{r_2} \alpha^{d_2}$ to the bridge node.
- The bridge node sends a challenge c, chosen at random, to the bidder node.
- If $m = 0$, the bidder sends $d_1, d_2 = c - d_1$ mod q, r_1, and $r_2 = w - r d_2$ mod q to the bridge node.
  If $m = 1$, the bidder sends $d_1 = c - d_2$ mod q , $d_2, r_1 = w - r d_1$ mod q, and r_2 to the bridge node
- The bridge node checks whether $c = d_1 + d_2$, mod q $a_1 = g^{r_1} \beta^{d_1}, b_1 = y^{r_1} (\alpha/g)^{d_1}, a_2 = g^{r_2} \beta^{d_2}$, and $b_2 = y^{r_2} \alpha^{d_2}$.

If all the equalities hold, the bidder is able to prove that the ciphertext decrypts to either 1 or 0.

3.4.3 Distributed Hash Table Overlay

Chord [66] is a distributed lookup protocol for mapping (and retrieving) a given key (or data) onto a specific node in a distributed and scalable manner. In particular, each node is assigned k-bit identifier, called "Chord ID", using *consistent hashing* [40]. The nodes' identifiers are arranged in the form of a modulo-2^k one-dimensional identifier circle known as *chord ring*. Each key is assigned to a peer node whose Chord ID is equal to or immediately next to the hash value of the key. In order to provide an efficient and scalable lookup service for the key, each node in the system stores information about a small fraction (i.e., $\log n$ where n is the network size) of all the nodes in what is known as the *finger table*. The i^{th} $(0 \leq i \leq (\log n - 1))$ element in the finger table of a node contains the identifier and the address (IP address, port number, etc.) of the node which is at 2^i distance apart in the chord ring. Figure 3.1 shows the finger tables of nodes u_0, u_1, and u_4. The outgoing arrows from a node in Figure 3.1 go to the nodes which are included in the current node's finger table. In order to look up a given key (or data) in the network, the source node hashes the key and searches in its finger tables. If it matches the Chord ID of certain node in its table, the node forwards the request to that node. Otherwise, it forwards the request to the node in its finger table which is closest to the hash value of the key in the chord ring. The same procedure follows at the new node until the request reaches the node which has the key. Both the lookup communication cost and the storage cost scale

as $O(\log n)$. In this study, we adapt the Chord DHT to design an efficient advertisement distribution scheme (Section 3.5.2.1) utilizing both DHT and social links.

3.5 A Secure and Private Auction Framework: SPA

The proposed secure and private auction framework for DOSNs, named SPA, consists of three phases. In the first phase called "*Identity Initiation*", all nodes that would like to participate in auctions obtain public/private pseudo IDs from a a Trusted Third Party (TTP). In the second phase called "*Buyer-Seller Matching*", the nodes that are interested in buying/selling item(s) distribute their intents through the social network via both DHT and social links during a specific time interval. For a particular item F_i[1] at a specific time interval T_k, a node, called *the bridge node* $v_{B^{i,k}}$, is chosen to match the sellers and buyers. In the third phase called "*Private Auction*", the bidders send sealed (encrypted) bids to the bridge node and the bridge node helps execute the auction. In what follows, we detail these three phases respectively.

3.5.1 Phase I: Identity Initiation

In order to be able to participate in an auction while preserving the ID privacy, each node requests a pair of public/private pseudo IDs from a TTP. Specifically, the TTP chooses two large primes $\tilde{p}$ and $\tilde{q}$ and publishes $N = \tilde{p}\tilde{q}$ along with a generator $\tilde{g}$ of a sufficiently large subgroup of Z_N^*. When a user v_i needs a public pseudo ID, it sends a signed request together with its certified public key to the TTP. For each such request, the TTP chooses a random ρ_i such that $gcd(\rho_i, \lambda(N)) = 1$, where $\lambda(\cdot)$ is the Carmichael function [55].

[1]The item F_i is drawn from a universal set of items or goods $\mathbb{F}$, $F_i \in \mathbb{F}$, such that same name is used for an item by all the participants in an auction.

ρ_i works as a public pseudo ID for user v_i. The TTP also computes a private pseudo ID $s_i = \tilde{g}^{1/\rho_i} = \tilde{g}^{d_i}$, where $\rho_i d_i = 1 \bmod \phi(N)$ and $\phi(N) = (p-1)(q-1)$ is the Eular's totient function. Note that all the calculations are modulo-N unless specified otherwise. The TTP signs the public pseudo ID, i.e., $S_i = K_D^{TTP}(\rho_i)$, and generates a certificate $C_i = (S_i, \rho_i)$, where K_D^{TTP} is the private key of the TTP. The private pseudo ID (s_i) and the certificate (C_i) are then securely delivered (encrypted with user v_i's public key) to user v_i.

3.5.2 Phase II: Buyer-Seller Matching

3.5.2.1 Advertisement Distribution

The absence of a central server in a DOSN necessitates the design of a distributed scheme to ensure the advertisement of a seller/buyer of an item reaches the right potential buyers/sellers. One naive solution is to broadcast the advertisements of sellers/buyers throughout the network. However, this will cause a serious message flooding in the network. For example, in a DOSN if each user has 100 friends on average, then the number of broadcast messages from a single user can, in the worst case, lead to 100 million messages in just 4 hops. This kind of broadcast flooding will inevitably congest the network and is not suitable for any practical application in a large-scale DOSN.

In contrast, SPA features a distributed advertisement distribution algorithm. Specifically, the sellers and the potential buyers distribute their intents through unicast advertisement messages utilizing both DHT and social links. All the advertisement messages for an item F_i in a time interval T_k intersect at the same bridge node $v_{B^{i,k}}$. The format of advertisement messages is shown in Table 3.1. $MessageID$ is a random ID chosen by the

source of the advertisement (a buyer or a seller), $SrcID$ changes in every hop and is the Chord ID of the current node, $H(\cdot)$ is a public hash function, B/S denotes whether the source is a buyer or a seller, and the $Payload$ of a seller's advertisement message contains the details about the item F_i, such as the price vector defined by the seller (see Section 3.5.3 for details) and others like shipping information/estimates. The bridge node is determined based on the hash value $a_{ik} = H(F_i||T_k)$. Particularly, the node $v_{B^{i,k}}$, whose Chord ID is either equal to or immediately next (in clockwise, i.e., increasing, order) to a_{ik} in the chord ring, serves as the bridge node for the item F_i during the time interval T_k. The time interval T_k is the time period, e.g., a day or a week, during which the auction for the item takes place. This timestamp T_k serves two important purposes: first, it puts a time limit on each trade, and more importantly, second, it randomly changes the bridge node in each time interval so that not a single node has to carry the computation overhead of being a bridge node all the time.

Table 3.1

The format of advertisement message.

$MessageID$	$SrcID$	$H(F_i\|\|T_k)$	B/S	$Payload$

The advertisement messages can be delivered to the bridge node as follows. Note that in addition to a finger table, we let each node keep the Chord ID and the address (IP address, port number, etc.) of its predecessor and of its successor on the Chord ring, as well as those of its friends and their predecessors on the Chord ring. When receiving an

$a_{ik} \leftarrow H(F_i||T_k)$;
$v_c \leftarrow$ Current Node ID ;
if $a_{ik} \in (predecessor(v_c), v_c]$ **then**
    Send an ACK message back to $SrcID$ and quit;
    /* v_c is the bridge node */
end
Store $MessageID, SrcID$ pair;
$SrcID \leftarrow v_c$;
if $(a_{ik} \in (v_c, sucessor(v_c)])$ **then**
    Forward advertisement message to $sucessor(v_c)$ and quit;
    /*$sucessor(v_c)$ is the bridge node*/
end
for $(\forall j | \exists e_{cj} \in E)$ **do**
    if $(a_{ik} \in (predecessor(v_j), v_j])$ **then**
        Forward the advertisement packet to v_j and quit ;
        /* Friend v_j of v_c is the bridge node */
    end
end
for $(j = 2 \rightarrow m)$ **do**
    if $(a_{ik} = j.finger(v_c))$ **then**
        Forward the advertisement packet to $j.finger(v_s)$ and quit;
        /* $j.finger(v_c)$ is the bridge node */
    end
end
$v_{next} \leftarrow \varnothing$;
for $(j = 1 \rightarrow m - 1)$ **do**
    if $a_{ik} \in (j.finger[v_c], (j+1).finger[v_c])$ **then**
        $v_{next} \leftarrow j.finger[v_c]$;
    end
end
if $(v_{next} = \varnothing)$ **then**
    $v_{next} \leftarrow m.finger[v_c]$;
end
for $(\forall j | \exists e_{cj} \in E)$ **do**
    if $(0 < (a_{ik} - v_j) < (a_{ik} - v_{next}))$ **then**
        $v_{next} \leftarrow v_j$;
    end
end
Forward the Advertisement Packet to v_{next}.

Figure 3.2

Algorithm 1: Distributed Advertisement Distribution.

advertisement message, each node first checks to see if it itself is the bridge node, i.e., if a_{ik} is equal to its Chord ID or between its predecessor's Chord ID and its Chord ID. If so, this node sends an acknowledgement message back to the sender of this advertisement message. Otherwise, it stores the $MessageID$ and $SrcID$ (along with the IP address of $SrcID$) of this message. The node then checks if its successor on the Chord ring is the bridge node, i.e., if a_{ik} is equal to its successor's Chord ID or between its own Chord ID and its successor's Chord ID. If so, it forwards the message to its successor. Otherwise, the node checks if any of the nodes in its finger table is the bridge node, i.e., if a_{ik} is equal to any node's Chord ID. If so, it forwards the message to that node. Otherwise, the node checks if any of its friends is the bridge nodes, i.e., if a_{ik} is equal to any friend's Chord ID or between any friend's Chord ID and its predecessor's Chord ID. If so, it forwards the message to that friend. Otherwise, it forwards the message to the node that precedes and is closest to a_{ik}, utilizing both its finger table and its friends list. The procedure continues until the message reaches the bridge node. The above advertisement distribution scheme is detailed in Algorithm 1 in Figure 3.2.

For example, as shown in Figure 3.1, assume that v_0 is the bridge node for an item F_i in time interval T_k, i.e., $a_{ik} \in (predecessor(v_0), v_0]$ where $predecessor(v_0)$ is the Chord ID of the predecessor of v_0 on the Chord ring. Suppose that node v_1 currently has the advertisement message. We can see that node v_1, its successor v_2, and the nodes in its finger table, and its friends are all not the bridge node. In this case, node v_1 forwards the message to the closest preceding node to the bridge node in the Chord ring, considering all the nodes in the finger table (v_2, v_3, v_5) and on its friend list (v_2, v_4, v_7), i.e., node v_7

in this example. Node v_7 then finds that its successor, i.e., node v_0, is the bridge node, and forwards the advertisement message to it. Ultimately, the bridge node receives the advertisement messages from all interested participants (seller and buyers) and starts the matching process.

3.5.2.2 Acknowledgement (ACK) Message Distribution

Every time a bridge node receives an advertisement message, it sends an ACK message back to the source node of the message reversely along the route that the message was delivered on. The format of ACK messages are shown in Table 3.2. In particular, each node on the route kept a record of $MessageID$ and $SrcID$ while forwarding the advertisement message to the bridge node. When forwarding the ACK message, the $DestID$ field is set to $SrcID$ of the corresponding advertisement message. The payload of the ACK message contains the address (e.g., IP address and the port number), of the bridge node where the price vectors of all the sellers are accessible and all the auction related computation (see Section 3.5.3) takes place under the scrutiny of all the bidders in the future. Once a buyer/seller node receives the ACK message, it connects to the bridge node via the Internet layer [2] to access the information provided by the sellers in their advertisement messages, and decides which one of the many sellers' items it wants to bid for.

[2]One may argue that the bridge node is able to know the IP address and may be able to identify the bidders when they connect to its page. However, the bidders can use services like Tor [21] to hide their true IP addresses.

Table 3.2

The format of ACK message.

$MessageID$	$DestID$	$H(F_i\|\|T_k)$	$Payload$

3.5.3 Phase III: Private Auction

Recall that in a DOSN, there are no trusted central auctioneers. Thus, distributed bidder-resolved auctions are indispensable for security and privacy purposes, in which bidders use cryptographic protocols to jointly determine the auction result. Previously proposed such auction schemes like [7] are not fully private and have high communication and computation complexities, which limit their usage in practical applications. In the following, we develop a private and more efficient auction protocol.

3.5.3.1 Outline

We first present the conceptual outline of the proposed private Vickrey based auction protocol. Without loss of generality, let us assume that a seller defines a price vector $\mathbf{p} = \begin{pmatrix} p_K p_{K-1} ... p_1 \end{pmatrix}^\intercal$ of K possible bidding prices. A bidder, say node v_i, submits a bid $\mathbf{b}^i = \begin{pmatrix} b^i_K b^i_{K-1} ... b^i_1 \end{pmatrix}^\intercal$, where $b^i_k \in \{0, 1\}$ and $1 \leq k \leq K$. If bidder v_i's bidding price is p_{l_i} $(1 \leq l_i \leq K)$, then b^i_k is equal to 1 when $k = l_i$ and equal to 0 otherwise.

We then define "*a doubly-integrated bid vector*", denoted by $\hat{\mathbf{b}}^{\mathbf{i}}$, for bidder v_i as

$$\hat{\mathbf{b}}^{\mathbf{i}} = \begin{pmatrix} \hat{b}^i_K \\ \hat{b}^i_{K-1} \\ \hat{b}^i_{K-2} \\ \vdots \\ \hat{b}^i_k \\ \vdots \\ \hat{b}^i_2 \\ \hat{b}^i_1 \end{pmatrix} = \begin{pmatrix} b^i_K \\ 2b^i_K + b^i_{K-1} \\ 2b^i_K + 2b^i_{K-1} + b^i_{K-2} \\ \vdots \\ 2b^i_K + 2b^i_{K-1} + \cdots + 2b^i_{k+1} + b^i_k \\ \vdots \\ 2b^i_K + 2b^i_{K-1} + \cdots + 2b^i_3 + b^i_2 \\ 2b^i_K + 2b^i_{K-1} + \cdots + 2b^i_2 + b^i_1 \end{pmatrix} \tag{3.1}$$

Thus, when the bidding price is p_{l_i} $(1 \leq l_i \leq K)$, the vector $\hat{\mathbf{b}}^{\mathbf{i}}$ is as follows:

$$\hat{b^i_k} = b^i_K \text{ when } k = K, \tag{3.2}$$

and

$$\hat{b^i_k} = \left(2 \sum_{m=k+1}^{K} b^i_m + b^i_k\right) = \begin{cases} 0, & \text{when } l_i < k < K, \\ 1, & \text{when } k = l_i \\ 2, & \text{when } k < l_i \end{cases}. \tag{3.3}$$

Assume that there are totally n bidders bidding for the same item. The sum of all the doubly-integrated bid vectors, denoted by $\hat{\mathbf{B}}$, can be obtained as[3]

$$\hat{\mathbf{B}} = \sum_{i=1}^{n} \hat{\mathbf{b}}^i. \tag{3.4}$$

The vector $\hat{\mathbf{B}}$'s elements would be 1, 3, 5, ..., $(2M-1)$ corresponding to the 1^{st}, 2^{nd}, 3^{rd}, ..., and M^{th} highest bidding prices.

[3]The bridge node can index the bidders by the order of received bids.

The **winning price calculation** process is described in the following. We calculate a vector $\boldsymbol{\mathcal{P}}$ as follows:

$$\boldsymbol{\mathcal{P}} = \left(\hat{\mathbf{B}} - 3 \cdot \mathbf{U}_K\right) * \mathbf{R} \tag{3.5}$$

where $\mathbf{R}$ is a random K-dimensional vector jointly generated by all the bidders ($\mathbf{R}(k) \neq 0$ for $1 \leq k \leq K$), $\mathbf{U_K}$ is a K-dimensional vector whose elements are all 1's, and '$*$' refers to component-wise multiplication. Thus, all the elements in $\boldsymbol{\mathcal{P}}$ are non-zero random numbers, except the element corresponding to the second highest bidding price which is zero. Thus, if $\boldsymbol{\mathcal{P}}(w) = 0$, then p_w is the winning price of the auction.

Winner bidder determination process follows the winning price calculation process. If a malicious winning bidder does not come forward and claim the bid, the auction would be incomplete and the item remains unsold. Therefore, in order to ensure non-repudiation, it is necessary to identify the winning bidder. Particularly, the winner of the auction is bidder v_i if $\mathcal{W}^i$ is zero, where

$$\mathcal{W}^i = (\hat{b}^i_w - 2) \cdot R_i \tag{3.6}$$

and R_i is a non-zero random number generated by bidder v_i. Example 1 shows an example for 4 bidders, in which X represents non-zero random values.

3.5.3.2 Cryptographic Protocol Design

Next we describe the details of the proposed cryptographic private auction protocol. Recall that after each bidder bidding for the same item receives an ACK message from the bridge node containing its address (IP address, port number, etc.), each bidder can access the advertisements from all the sellers available at the bridge node and decide which par-

Example 1: Suppose that the price vector given by a seller is $\mathbf{p} = \begin{pmatrix}150140130120110100\end{pmatrix}^\intercal$. Assume that there are four bidders: v_1, v_2, v_3, and v_4, and their bidding prices are 140, 130, 120, and 110, respectively. Therefore, $\mathbf{b}^1 = \begin{pmatrix}010000\end{pmatrix}^\intercal$, $\mathbf{b}^2 = \begin{pmatrix}001000\end{pmatrix}^\intercal$, $\mathbf{b}^3 = \begin{pmatrix}000100\end{pmatrix}^\intercal$, $\mathbf{b}^4 = \begin{pmatrix}000010\end{pmatrix}^\intercal$, Then, we have

$$\hat{\mathbf{B}} = \hat{\mathbf{b}}^1 + \hat{\mathbf{b}}^2 + \hat{\mathbf{b}}^3 + \hat{\mathbf{b}}^4 = \begin{pmatrix}0\\1\\2\\2\\2\\2\end{pmatrix} + \begin{pmatrix}0\\0\\1\\2\\2\\2\end{pmatrix} + \begin{pmatrix}0\\0\\0\\1\\2\\2\end{pmatrix} + \begin{pmatrix}0\\0\\0\\0\\1\\2\end{pmatrix} = \begin{pmatrix}0\\1\\3\\5\\7\\8\end{pmatrix}, \text{ and } \boldsymbol{\mathcal{P}} = \begin{pmatrix}0\\1\\3\\5\\7\\8\end{pmatrix} - \begin{pmatrix}3\\3\\3\\3\\3\\3\end{pmatrix} \cdot \begin{pmatrix}\mathbf{R}(1)\\\mathbf{R}(2)\\\mathbf{R}(3)\\\mathbf{R}(4)\\\mathbf{R}(5)\\\mathbf{R}(6)\end{pmatrix} = \begin{pmatrix}X\\X\\0\\X\\X\\X\end{pmatrix}.$$

Since we have $\boldsymbol{\mathcal{P}}(3) = 0$, the winning price is $p_w = p_3 = 130$. According to (3.6), we get $\mathcal{W}_3^1 = (\hat{b}_3^1 - 2) \cdot R_1 = (2-2) \cdot R_1 = 0$, $\mathcal{W}_3^2 = (\hat{b}_3^2 - 2) \cdot R_2 = (1-2) \cdot R_2 = X$, $\mathcal{W}_3^3 = (\hat{b}_3^3 - 2) \cdot R_3 = (0-2) \cdot R_3 = X$, $\mathcal{W}_3^4 = (\hat{b}_3^4 - 2) \cdot R_4 = (0-2) \cdot R_4 = X$. Thus, the winning bidder is v_1.

ticular seller's item to bid for. The bidders then send to the bridge node their encrypted bids according to the price vector defined by the seller they choose. The bridge node finally determines the winning price and the winning bidder. The proposed auction protocol consists of five processes as follows.

Public Pseudo ID Authentication is the first process of the proposed auction protocol. After receiving a buying advertisement message from a bidder, the bridge node needs to verify if it is an authentic user in the network so as to defend attacks like impersonation. Thus, a bidder node v_i needs to prove that it possesses the private pseudo ID s_i corresponding to the pubic pseudo ID ρ_i. We apply Fiat-Shamir heuristic to convert the interactive proof [59] between a prover (a bidder) and a verifier (the bridge node) to a non-interactive proof. Note that the purpose of having non-interactive zero knowledge (NIZK) proofs is not only to reduce the communication complexity between the bidders and the bridge node, but more importantly, to relax the assumption on trustworthy bridge nodes (i.e., honest verifiers) in ZKPs. This is because the non-interactive proof of authenticity can be verified by all the parties participating in the auction and a dishonest bridge node will get caught. In particular, the Fiat-Shamir heuristic [25] makes use of a hash function $\tilde{H}(\cdot)$, modelled as a Random Oracle (RO), to construct a random challenge from the verifier. The public pseudo ID authentication can be carried out following the steps below:

- Bidder v_i chooses a random $\tilde{r}$, calculates $z = \tilde{r}^{\rho_i} \mod N$, and sends $z, y = \tilde{r} {s_i}^c \mod N$, and the certificate C_i to the bridge node, where $c = \tilde{H}(z)$.
- The bridge node checks and accepts the proof if $z = y^{\rho_i} \tilde{g}^{-c} \mod N$.

Theorem 4

A legal node can always be successfully authenticated.

Proof: Note that the bridge node can obtain the public pseudo ID ρ_i from the certificate C_i and that $s_i = \tilde{g}^{1/\rho_i} \mod N$. Thus, we have

$$y^{\rho_i}\tilde{g}^{-c} \equiv (\tilde{r}s_i^c)^{\rho_i}\tilde{g}^{-c} \equiv \tilde{r}^{\rho_i}\tilde{g}^c\tilde{g}^{-c} \equiv \tilde{r}^{\rho_i} \equiv z \mod N. \tag{3.7}$$

■

Theorem 5

[Soundness] An illegal bidder node, who does not have a valid s_i and can not compute in polynomial time the ρ_i-th root, has only negligible probability of begin successfully authenticated.

Proof:An illegal bidder may be able to deceive the bridge node (verifier) if $\tilde{r}+c$ is divisible by v_i and sends $z = \tilde{g}^r \mod N$ and $y = \tilde{g}^{(\tilde{r}+c)/\rho_i} \mod N$. The bridge node will accept the proof, because

$$y^{\rho_i}\tilde{g}^{-c} \equiv (\tilde{g}^{(\tilde{r}+c)/\rho_i})^{\rho_i} \equiv \tilde{g}^{\tilde{r}+c}\,\tilde{g}^{-c} \equiv z \mod N. \tag{3.8}$$

However, the probability of this event is very low ($\sim 1/N$, where N is a very large number, e.g, 1024 bit number)

Next we prove by contradiction [32] that an illegal bidder, without a valid s_i, cannot increase this probability. In order to increase the probability, assume first that the bidder is able to compute ρ_i-th roots y' and y'' of $z\tilde{g}^c$ for two challenges c' and c''. Choose Bezout coefficients $\tilde{m}$ and $\tilde{k}$ such that:

$$\rho_i\tilde{m} + (c' - c'')\tilde{k} = \pm 1 \tag{3.9}$$

We have, $\gcd(\rho_i, c' - c'') = 1$, therefore, there always exist Bezout coefficients $\tilde{m}$ and $\tilde{k}$. The following computation reveals s_i

$$\begin{aligned}\tilde{g}^m \left(\frac{y'}{y''}\right)^{\tilde{k}}\right)^{\pm 1} &\equiv s_i^{\rho_i \tilde{m}} \left(\left(\frac{y'}{y''}\right)^{\tilde{k}}\right)^{\pm 1} \\ &\equiv (s_i^{\rho_i \tilde{m}} s_i^{(c'-c'')\tilde{k}})^{\pm 1} \equiv s_i \bmod n \end{aligned} \tag{3.10}$$

This, however, contradicts with the assumption that the bidder does not know s_i corresponding to ρ_i.

■

Note that in order to further reduce the communication cost between the bidders and the bridge node, bidders can include this non-interactive proof in the payload of their advertisement messages as mentioned before.

Distributed Encryption Key Generation process follows public pseudo ID authentication process. Each bidder then chooses a random key $x_i \in \mathbb{G}_q$ and sends $y_i = g^{x_i} \mod p$ to the bridge node with a ZKP of the knowledge of x_i, i.e., a discrete logarithm regarding y_i (Section 3.4.2.1). The bridge node makes all the y_i's and the corresponding ZKPs public. Each bidder can compute the encryption key (public key) as $y = \prod_{i=1}^{n} y_i$. Note that similarly, in order to reduce the communication complexity of interactive ZKPs and relaxing the assumption on a reliable bridge node, we employ Fiat-Shamir heuristic [25] to make the ZKP (and all the following ZKPs as well) non-interactive, i.e., use NIZK proofs.

In the **Bid Encryption** process, each bidder prepares his/her own bid and sends the encrypted bid to the bridge node as follows.

- *Bid Preparation:* Without loss of generality, we denote a seller's price vector by $\mathbf{p} = \left(p_K p_{K-1} \cdots p_1\right)^{\intercal}$ and a bidder's (node v_i's) bidding vector by $\mathbf{b}^i = \left(b^i_K b^i_{K-1} ... b^i_1\right)^{\intercal}$.

Suppose node v_i's bidding price is p_{l_i}. Then, we have b_k^i $(1 \leq k \leq K)$ is equal to 1 when $k = l_i$ and equal to 0 otherwise.

- *Bid Encryption:* The bidder then encrypts the bidding vector with the encryption (public) key element by element, i.e., for any $1 \leq k \leq K$, the bidder computes $Enc(b_k^i) = \langle \alpha_k^i, \beta_k^i \rangle = \langle g^{r_k^i}, g^{b_k^i} y^{r_k^i} \rangle$ where $r_k^i \in \mathbb{G}_q$ is a random number generated by bidder v_i.

  ZKP Generation: The bidder v_i needs to prove that the encrypted bidding vector is generated adhering to the protocol. In particular, it needs to prove the following facts in zero knowledge:

  - *Each element in its bidding vector is the encryption of either 1 or 0.* The bidder generates a ZKP as described in Section 3.4.2.3.
  - *Only one element in its bid vector corresponds to 1, i.e.,* $\sum_{k=1}^{K} b_k^i = 1$. The bridge node uses the protocol described in 3.4.2.2 to show $\log_y \left(\frac{\prod_{k=1}^{K} \beta_k^i}{g} \right) = \log_g (\prod_{k=1}^{K} \alpha_k x_i)$ in zero knowledge.

- *Bid Signing and Publishing:* Note that the encrypted bidding vectors obtained above are repudiable. Before sending the encrypted bids to the bridge node, in order to ensure authentication and non-repudiation, all bidders sign their bids with an anonymous (pseudo ID based) signature scheme [59] shown below. In the following, we detail the process for bidder v_i to sign each of the encrypted elements in the bidding vector $\mathbf{b}^i$, i.e., $Enc(b_k^i) = \langle \alpha_k^i, \beta_k^i \rangle = \langle g^{r_k^i}, g^{b_k^i} y^{r_k^i} \rangle$, and for the bridge node to verify it. The calculations in this process take place in modulo-N unless mentioned otherwise.

  - Bidder v_i computes $z_{\alpha_k^i} = r_{\alpha_k^i}^{\rho_i}$, $\epsilon_{\alpha_k^i} = h(\alpha_k^i || z_{\alpha_k^i})$, and $y_{\alpha_k^i} = r_{\alpha_k^i} s_i^{\epsilon_{\alpha_k^i}}$, where $r_{\alpha_k^i}$ is a random number generated by v_i, and $h(\cdot)$ is a publicly known hash function. Bidder v_i also computes $\epsilon_{\beta_k^i}$ and $y_{\beta_k^i}$ in a similar way. Then, bidder v_i generates the signature for the encrypted bid $Enc(b_k^i) = \langle \alpha_k^i, \beta_k^i \rangle$, which is $\langle (\epsilon_{\alpha_k^i}, y_{\alpha_k^i}), (\epsilon_{\beta_k^i}, y_{\beta_k^i}) \rangle$, and sends it along with his/her certificate C_i to the bridge node.
  - The bridge node obtains the public pseudo ID ρ_i of bidder v_i from the certificate C_i and computes $m_{\alpha_k^i} = y_{\alpha_k^i}^{\rho_i} \tilde{g}^{-\epsilon_{\alpha_k^i}}$ and $m_{\beta_k^i} = y_{\beta_k^i}^{\rho_i} \tilde{g}^{-\epsilon_{\beta_k^i}}$. The bridge node accepts the bid if $h(\alpha_k^i || m_{\alpha_k^i}) = \epsilon_{\alpha_k^i}$ and $h(\beta_k^i || m_{\beta_k^i}) = \epsilon_{\beta_k^i}$.

Theorem 6

If the bid from bidder v_i is authentic, the following verification equations would hold:

$h(\alpha_k^i || m_{\alpha_k^i}) = \epsilon_{\alpha_k^i}$ *and* $h(\beta_k^i || m_{\beta_k^i}) = \epsilon_{\beta_k^i}$ *for any* $1 \leq k \leq K$.

Proof: We present the proof by dropping the superscripts/subscripts of the subscripts in the notations above for simplicity. Particularly, since $m_\alpha = y_\alpha^{\rho_i}\tilde{g}^{-\epsilon_\alpha} = (r_\alpha s_i^{\epsilon_\alpha})^{\rho_i}\tilde{g}^{-\epsilon_\alpha} = r_\alpha^{\rho_i}(s_i^{\rho_i})^{\epsilon_\alpha}\tilde{g}^{-\epsilon_\alpha} = r_\alpha^{\rho_i}\tilde{g}^{\epsilon_\alpha}\tilde{g}^{-\epsilon_\alpha} = r_\alpha^{\rho_i} = z_\alpha$ (note that $s_i^{\rho_i} = \tilde{g}^{d_i\rho_i} = \tilde{g}$), we have $h(\alpha||m_\alpha) = h(\alpha||z_\alpha) = \epsilon_\alpha$. Similarly, we can prove that $m_\beta = z_\beta$ and hence $h(\beta||m_\beta) = h(\beta||z_\beta) = \epsilon_\beta$. ■

Note that any participants in the auction can check the verification equations.

Theorem 7

[Soundness] An illegal bidder node, who generates signature without valid s_i, has negligible probability of success for signed bid verification by a bridge node.

Proof: A bidder node v_i signs his/her bid vector ,i.e, $Enc(b_k^i) = \langle \alpha_k^i, \beta_k^i \rangle, 1 \leq k \leq K$, with the signature scheme. For each element $Enc(b_k^i) = \langle \alpha_k^i, \beta_k^i \rangle$, in the vector, one can see from theorem 5, that if $(r_{\alpha_k^i}^{\rho_i} + \epsilon_{\alpha_k^i})$ and $(r_{\beta_k^i}^{\rho_i} + \epsilon_{\beta_k^i})$ are divisible by ρ_i, a malicious bidder is able to manipulate $z_{\alpha_k^i}, z_{\beta_k^i}$ and $y_{\alpha_k^i}, y_{\beta_k^i}$ to successfully convince the verifier (bridge node) node the authenticity of the signature on the bid element. However, the probability of this event is $< 1/N$ and the probability of such events for the whole bid vector is $\ll 1/N$ and is negligible. Similarly, following the proof in theorem 5, a malicious bidder is unable to increase this probability. Hence a signature generated by an illegal bidder without valid s_i has negligible probability of being successfully verified by the bridge node. ■

Winning Price Determination process is executed to calculate the auction clearing price. Once all the bids are received within the time frame of current auction, the bridge

node combines the encrypted bidding vectors to obtain the encrypted doubly-integrated bid vector. For each bidder v_i ($1 \leq i \leq n$), the bridge node computes $Enc(\hat{\mathbf{b}^i})$ as:

$$Enc(\hat{\mathbf{b}}^i) = \left(\langle \hat{\alpha}_K^i, \hat{\beta}_K^i \rangle \langle \hat{\alpha}_{K-1}^i, \hat{\beta}_{K-1}^i \rangle ... \langle \hat{\alpha}_1^i, \hat{\beta}_1^i \rangle \right)^\intercal \tag{3.11}$$

where $\langle \hat{\alpha}_k^i, \hat{\beta}_k^i \rangle$ is

$$\begin{aligned} &\left\langle \prod_{m=k+1}^{K} (\alpha_m^i)^2 \cdot \alpha_k^i, \prod_{m=k+1}^{K} (\beta_m^i)^2 \cdot \beta_k^i \right\rangle \left(\right. \\ =&\left\langle g^{\left(\sum_{m=k+1}^{K} 2r_m^i + r_k^i \right.}, g^{\sum_{m=k+1}^{K} 2b_m^i + b_k^i} y^{\left. \sum_{m=k+1}^{K} 2r_m^i + r_k^i \right)} \right\rangle \end{aligned} \tag{3.12}$$

when $1 \leq k < K$, and $\langle g^{r_k^i}, g^{b_k^i} y^{r_k^i} \rangle$ when $k = K$. Define δ_k^i as $\sum_{m=k+1}^{K} 2r_m^i + r_k^i$ when $1 \leq k < K$ and r_k^i when $k = K$. Thus, for any $1 \leq k \leq K$, we have

$$\langle \hat{\alpha}_k^i, \hat{\beta}_k^i \rangle = \langle g^{\delta_k^i},\ g^{\hat{b_k^i}} y^{\delta_k^i} \rangle \tag{3.13}$$

where $\hat{b_k^i}$ is defined in (3.2) and (3.3).

Similarly, the bridge node can obtain the encryption of the sum of all the doubly-integrated bid vectors as follows

$$\begin{aligned} & Enc(\hat{\mathbf{B}}) \\ = & \left(\prod_{i=1}^{n} Enc(\hat{\mathbf{b}}_K^i) \quad \dots \quad \prod_{i=1}^{n} Enc(\hat{\mathbf{b}}_1^i) \right)^\intercal \\ = & \left(\langle \prod_{i=1}^{n} \hat{\alpha_K^i}, \prod_{i=1}^{n} \hat{\beta_K^i} \rangle \quad \dots \quad \langle \prod_{i=1}^{n} \hat{\alpha_1^i}, \prod_{i=1}^{n} \hat{\beta_1^i} \rangle \right)^\intercal \\ = & \left(\langle \hat{\alpha}_{B_K}, \hat{\beta}_{B_K} \rangle \quad \langle \hat{\alpha}_{B_{K-1}}, \hat{\beta}_{B_{K-1}} \rangle \quad \dots \quad \langle \hat{\alpha}_{B_1}, \hat{\beta}_{B_1} \rangle \right)^\intercal \end{aligned} \tag{3.14}$$

where for any $1 \leq k \leq K$,

$$\langle \hat{\alpha}_{B_k}, \hat{\beta}_{B_k} \rangle = \left\langle g^{\left(\sum_{i=1}^{n} \delta_k^i \right.}, g^{\sum_{i=1}^{n} \hat{b}_k^i} y^{\left. \sum_{i=1}^{n} \delta_k^i \right)} \right\rangle \left(\right. \tag{3.15}$$

Recall that we determine the winning price through (3.5). Thus, the bridge node first computes the encryption of a vector $\mathbf{P} = \hat{\mathbf{B}} - 3 \cdot \mathbf{U}_K$ as follows:

$$Enc(\mathbf{P}) = Enc(\hat{\mathbf{B}} - 3 \cdot \mathbf{U}_K) = Enc(\hat{\mathbf{B}}) * Enc(-3\mathbf{U}_K) \tag{3.16}$$

which we denote by

$$\left(\langle \alpha_{P_K}, \beta_{P_K} \rangle \quad \langle \alpha_{P_{K-1}}, \beta_{P_{K-1}} \rangle \quad \ldots \quad \langle \alpha_{P_1}, \beta_{P_1} \rangle \right)^{\intercal}.$$

The bridge node then publishes the above calculations on its public profile, so that all the bidders can verify the correctness of the computations.

In the next step, each bidder participates in the distributed decryption of the clearing price. Specifically, each bidder v_i computes and sends to the bridge node

$$\begin{aligned} \boldsymbol{\alpha}'_{P^i} &= \left((\alpha_{P_K})^{R^i_K} \quad (\alpha_{P_{K-1}})^{R^i_{K-1}} \quad \ldots \quad (\alpha_{P_1})^{R^i_1} \right)^{\intercal} \\ \boldsymbol{\beta}'_{P^i} &= \left((\beta_{P_K})^{R^i_K} \quad (\beta_{P_{K-1}})^{R^i_{K-1}} \quad \ldots \quad (\beta_{P_1})^{R^i_1} \right)^{\intercal} \end{aligned} \tag{3.17}$$

where $\mathbf{R}^i$ is a K-dimensional vector of non-zero random numbers generated by bidder v_i. In addition to $\boldsymbol{\alpha}'_{P^i}$ and $\boldsymbol{\beta}'_{P^i}$, each bidder also proves in zero knowledge that the corresponding elements, e.g., the kth elements, of the vectors $\boldsymbol{\alpha}'_{P^i}$ and $\boldsymbol{\beta}'_{P^i}$ are obtained using the same random value, e.g., R^i_k (as shown in Section 3.4.2.2).

The bridge node then combines the received values from the bidders to calculate $\langle \boldsymbol{\alpha}'_{\mathbf{P}}, \boldsymbol{\beta}'_{\mathbf{P}} \rangle$ as follows:

$$\langle \prod_{i=1}^{n} \boldsymbol{\alpha}'_{P^i}(K), \prod_{i=1}^{n} \boldsymbol{\beta}'_{P^i}(K) \rangle, \ldots, \langle \prod_{i=1}^{n} \boldsymbol{\alpha}'_{P^i}(1), \prod_{i=1}^{n} \boldsymbol{\beta}'_{P^i}(1) \rangle \Bigg)^{\intercal} \tag{3.18}$$

Thus, all the bidders can calculate the winning price of the auction by following the distributed encryption approach, i.e., computing

$$\mathbf{\Omega} = \left(\frac{\boldsymbol{\beta}'_{\mathbf{P}}(K)}{\prod_{i=1}^{n}(\boldsymbol{\alpha}'_{\mathbf{P}}(K))^{x_i}} \quad \frac{\boldsymbol{\beta}'_{\mathbf{P}}(K-1)}{\prod_{i=1}^{n}(\boldsymbol{\alpha}'_{\mathbf{P}}(K-1))^{x_i}} \quad \cdots \quad \frac{\boldsymbol{\beta}'_{\mathbf{P}}(1)}{\prod_{i=1}^{n}(\boldsymbol{\alpha}'_{\mathbf{P}}(1))^{x_i}} \right)^{\mathsf{T}} \tag{3.19}$$

Where $(\boldsymbol{\alpha}'_{\mathbf{P}}(k))^{x_i}$'s $(1 \leq k \leq K)$ are transmitted by bidder v_i to the bridge node and made public (along with a proof that the same x_i was used as in the distributed key generation process, as shown in Section 3.4.2.2), and

$$\begin{aligned} \Omega(k) &= \frac{\prod_{i=1}^{n} \boldsymbol{\beta}'_{P^i}(k)}{\prod_{i=1}^{n}(\prod_{i=1}^{n} \boldsymbol{\alpha}'_{P^i}(1))^{x_i}} = \frac{(\beta_{P_k})^{\sum_{i=1}^{n} R_k^i}}{(\alpha_{P_k})^{\sum_{i=1}^{n} R_k^i \sum_{i=1}^{n} x^i}} \\ &= \frac{(g^{\mathbf{P}(k)} \cdot y^r)^{\sum_{i=1}^{n} R_k^i}}{(g^r)^{\sum_{i=1}^{n} R_k^i \sum_{i=1}^{n} x^i}} = g^{\mathbf{P}(k) \cdot \sum_{i=1}^{n} R_k^i} = g^{\boldsymbol{\mathcal{P}}(k)}. \end{aligned} \tag{3.20}$$

where $\boldsymbol{\mathcal{P}}$ is defined in (3.5). Therefore, the element p_w of the price vector $\mathbf{p}$ is the winning price if $\Omega(w) = g^{\boldsymbol{\mathcal{P}}(w)} = g^0 = 1$.

Winning Bidder Identification process follows winning price calculation process and is needed to identify the winning bidder's pseudo ID. Recall that the winning bidder can be determined by checking if (3.6) is equal to 0. Only the winning bidder's public pseudo ID will be known to others. The winning bidder determination process is as follows.

First, for any $1 \leq i \leq n$, the bridge node computes,

$$Enc(W^i) = Enc(\hat{b}_w^i - 2) = Enc(\hat{b}_w^i) \cdot Enc(-2) \tag{3.21}$$

which we denote by $\langle \alpha_{W^i}, \beta_{W^i} \rangle$.

Then, each bidder v_i computes $\langle \alpha_{W^i}^{R_i}, \beta_{W^i}^{R_i} \rangle$, and sends it and a ZKP (as shown in Section 3.4.2.2) that $\alpha_{W^i}^{R_i}$, $\beta_{W^i}^{R_i}$ are computed using the same random number to the bridge node.

The bridge node makes such values public and ask all the bidders to jointly decrypt for W^i's. Particularly, for any W^i ($1 \leq i \leq n$), each bidder v_j transmits $(\alpha_{W^i}^{R_i})^{x_j}$ (along with a proof that these n x_j's are the same as that used as in the distributed key generation process, as shown in Section 3.4.2.2) to the bridge node, which can then compute

$$\Phi^i = \frac{\beta_{W^i}^{R_i}}{\prod_{j=1}^{n}(\alpha_{W^i}^{R_i})^{x_j}} = \frac{(g^{W^i} \cdot y^r)^{R_i}}{(g^r)^{R_i \sum_{j=1}^{n} x_j}} = g^{W^i R_i} = g^{\mathcal{W}^i}. \tag{3.22}$$

Finally, bidder v_i is the winning bidder if $\mathcal{W}^i = 0$, or $\Phi^i = g^0 = 1$.

3.5.3.3 Tie Breaking

We find that the auction scheme presented above fails to produce an outcome if there is a tie in the highest or/and the second highest bidding price. A simple solution would be to decrypt all the elements of vector $\hat{\mathbf{B}}$, which gives the locations of all the ties (including the ties in the highest bid and the second highest bid) and the winning price as well. However, revealing $\hat{\mathbf{B}}$ in public constitutes a breach in privacy of the bidders whose bidding statistics will be available to potential adversaries who can use the information to their advantage in future auctions. The only information needed to be revealed is the winning price and the pseudo public ID of the winning bidder. In the following, we develop a scheme to determine the auction result in presence of tie(s).

Particularly, notice that the vector $(\mathbf{B} - t \cdot \mathbf{U_K})$, where $\mathbf{B} = \sum_{i=1}^{n} \mathbf{b}^i$, results in 0 at each location corresponding to the element in the price vector where there is a tie of t bidders. Besides, the vector $\hat{\mathbf{B}} - (t+2h) \cdot \mathbf{U_K}$ leads to 0 at the second highest bid position if t bidders bid the same second highest price and h bidders bid the same highest price. Thus, if $(\mathbf{B} - t \cdot \mathbf{U_K})$ and $\hat{\mathbf{B}} - (t+2h) \cdot \mathbf{U_K}$ both result in 0 at the same location, then that

is corresponding to the second highest price in the price vector. Consequently, the bridge node can first calculate the following vector

$$\boldsymbol{\mathcal{P}}^{t,h} = \Big((\mathbf{B} - t \cdot \mathbf{U_K}) + (n+1)(\hat{\mathbf{B}} - (t+2h) \cdot \mathbf{U_K})\Big) * \mathbf{R} \tag{3.23}$$

for $1 \leq h \leq n-t$ and $1 \leq t \leq n-1$, where $\mathbf{R}$ is a K-dimensional nonzero random vector jointly generated by the bidders, and the second term is multiplied by $(n+1)$ to make sure the two terms do not accidently add up to zero. This vector can be re-written as

$$\boldsymbol{\mathcal{P}}^{t,h} = \big((\mathbf{B} + (n+1)\hat{\mathbf{B}}\big) - \big((n+2)t + 2(n+1)h)\mathbf{U_K}\big) * \mathbf{R} \tag{3.24}$$

and the winning price is p_w if $\boldsymbol{\mathcal{P}}^{t,h}(w) = 0$. In order to ensure security and privacy, we can follow the cryptographic process presented in Section 3.5.3.2 to verify if $g^{\boldsymbol{\mathcal{P}}^{t,h}(w)} = g^0 = 1$.

Similarly, winning bidders can be identified by checking if$(\hat{b}_w^i - 2) \cdot R_i = 0$ along the line in Section 3.5.3.2, where R_i is a nonzero random number generated by bidder v_i. In the case of a tie at the highest bidding price, some specific rules can be employed to determine the final winner, e.g., the bidder who submitted his/her bid first among all the winners.

3.5.3.4 (M+1)st Price Auction

The private auction scheme that we have developed so far is for the case where each seller has one unit of an item to sell at a time and a buyer is also interested in buying only one unit of the item at a time. In this part, we investigate private auction for the scenarios where a seller has multiple, say M ($M \geq 1$), units of the same items to sell, i.e., private $(M+1)$-st price auction. In particular, each buyer is interested in buying one unit of the

item and the top M bidders are winners who pay the $(M+1)$-st highest bidding price. Note that when $M = 1$, the $(M+1)$st-price auction reduces to the 2nd price (Vickrey) auction investigated above.

The basic idea for private $(M+1)$-st price auction is as follows. When there is no tie in the bidding prices, the winning price in an $(M+1)$-st price auction can be obtained by first calculating the vector below in a similar way to (3.5), i.e.,

$$\overline{\boldsymbol{\mathcal{P}}} = \left(\hat{\mathbf{B}} - (2M+1) \cdot \mathbf{u}_k\right) \left(* \mathbf{R}. \right. \tag{3.25}$$

The winning price is p_w if $\overline{\boldsymbol{\mathcal{P}}}(w) = 0$. Similarly, the winning bidders can be identified if

$$\overline{\mathcal{W}}^i = (\hat{b}_w^i - 2) \cdot R_i = 0. \tag{3.26}$$

When there are ties in the bidding prices, the winning price and the winning bidders can be determined by following the same approach in Section 3.5.3.3. The cryptographic process in Section 3.5.3.2 can be employed to provide security and privacy.

3.5.4 The Case for Bidders Dropping out Prematurely

The auction protocols defined above work requires the bidders, who participated in the public key computation process, continue in the private auction process until all the calculations regarding the winning price calculation and winning bidder determination are done. If one or more bidder do not participate in the combined result calculation, the process fails to yield any result and the auction needs to be conducted again. In order to deter bidder nodes from leaving the auction scheme before the final calculations, the bridge node can refer the misbehaving bidders to the TTP for penalty. The TTP may deny

to assign pseudo ID to such bidders in future and the current pseudo ID may be blacklisted and barred from future auctions by bridge nodes.

3.6 Performance Analysis

In this section we analyze the performance of the proposed secure and private auction protocol, in terms of computation and communication costs, and security and privacy.

3.6.1 Computation and Communication Costs

3.6.1.1 Computation Cost

In what follows, we analyse the computational complexity of each bidders and that of the bridge node, respectively.

A Bidder's Computation Complexity is analyzed in detail below. In the *pseudo ID authentication* process, a bidder conducts 2 exponentiations, denoted by EXP, in the NIZK proof. In the *distributed encryption key generation* process, a bidder carries out $1 \times EXP$ for public key generation and $1 \times EXP$ for the corresponding ZKP. In the *bid encryption* process, a bidder conducts $3K \times EXP$ for "bid encryption", $(6K+2) \times EXP$ for "ZKP generation", and $4K \times EXP$ for "bid signing and publishing". In the *winning price calculation* process, a bidder needs to compute $3K \times EXP$ and another $(3K + 1) \times EXP$ for the ZKPs. In the *winning bidder determination* process, a bidder needs to compute $(n + 2) \times EXP$ and another $(n + 2) \times EXP$ for the ZKPs. Therefore, the total computational complexity of a bidder is $(2n + 19K + 11) \times EXP$, i.e., on the order of $O(n + K)$ EXP operations. Note that we ignore the multiplication operation, denoted by MUL, as it is insignificant compared to exponentiations.

A Bridge Node's Computation Complexity calculation is shown below in detail. In the *pseudo ID authentication* process, a bridge node conducts $2 \times EXP$ and $1 \times MUL$ for each bidder, i.e., $2n \times EXP$ and $n \times MUL$ in total. In the *distributed encryption key generation* process, a bridge node carries out $2 \times EXP$ and $1 \times MUL$ in the ZKP for each bidder, and $(n-1) \times MUL$ for computing the encryption key, i.e., $2n \times EXP$ and $(2n-1) \times MUL$ in total. In the *bid encryption* process, a bridge node computes $8nK \times EXP$ and $4nK \times MUL$ for the first ZKP and $4n \times EXP$ and $2n \times MUL$ for the second ZKP for "ZKP generation", and $4nK \times EXP$ and $2nK \times MUL$ for "bid signing and publishing", i.e., $(12nK + 4n) \times EXP$ and $(6nK + 2n) \times MUL$ in total. In the *winning price calculation* process, a bridge node needs to conduct $6nK \times MUL$ for $Enc(\hat{\mathbf{b}}^i)$, $2nK \times MUL$ for $Ecn(\hat{\mathbf{B}})$, $3K \times EXP$ and $4K \times MUL$ for $Enc(\mathbf{P})$, $3nK \times MUL$ for $\mathbf{\Omega}$, and $(8nK+4n) \times EXP$ and $(4nK+2n) \times MUL$ for the two ZKPs, i.e., $(8nK + 4n + 3K) \times EXP$ and $(15nK + 2n + 4K) \times MUL$ in total. In the *winning bidder determination* process, a bridge node computes $6n \times EXP$ and $4n \times MUL$ in the first step, and $n^2 \times MUL$ in the second step, and $(4n^2+8n) \times EXP$ and $(2n^2+4n) \times MUL$ for the two ZKPs, i.e., $(4n^2 + 14n) \times EXP$ and $(3n^2 + 8n) \times MUL$ in total. Thus, the total computational complexity of a bridge node is $(4n^2 + 20nK + 24n + 3K) \times EXP$ and $(3n^2 + 21nK + 14n + 4K - 1) \times MUL$, i.e., on the order of $O(n^2 + nK)\ EXP$ and $O(n^2 + nK)\ MUL$ operations.

Note that most of the above computation cost for a bridge node is ZKP verification related computation costs (all the n^2,and nk terms for EXP operation) In our protocols, the bidder nodes provide non-interactive ZKPs that can be verified by any participants of

the auction process. bridge node can distribute the computation load for proof verifications (including those for ZKPs and for signatures) to other nodes without increasing computation and communication complexity for a bidder node. Besides, those ZKPs can be verified by any other participants in the auction. For example, bidder node j can verify the ZKPs of node $(j + k)\ mod\ n$, where $k \in [1, n - 1]$. In this case the communication complexity of all the bidder nodes and the bridge node will be $O(n + K)\ EXP$. It is also important to point out that [7] does not include the cost for ZKP verification and still has $O(nK)$ EXP complexity. We can see that our proposed protocol has much lower computational complexity.

3.6.1.2 Communication Cost

Note that the communication cost mainly comes from the bidders since all the ZKPs are non-interactive. We analyze the communication cost of each bidder as follows.

In the *pseudo ID authentication* process, a bidder transmits z, y, and C_i to the bridge node, i.e., $4\lceil \log N \rceil$ bits. In the *distributed key generation* process, a bidder sends one $\lceil \log p \rceil$ bits to construct the public key and one $\lceil \log p \rceil + \lceil \log q \rceil$ bits ZKP to the bridge node, i.e., $2\lceil \log p \rceil + \lceil \log q \rceil$ in total. In the *bid encryption* process, each bidder transmits $K((4\lceil \log p \rceil + 4\lceil \log q \rceil) + (2\lceil \log p \rceil + \lceil \log q \rceil))$ bits to prove the bids fulfill the given requirements for "ZKP generation". Each bidder also sends K ElGamal ciphertexts ($2K\lceil \log p \rceil$ bits), K corresponding signatures ($2K(\lceil \log N \rceil + |h|)$ bits with $|h|$ being the size of hash digest), and his/her certificate ($2\lceil \log N \rceil$ bits). So all the cost in this process is $8K\lceil \log p \rceil + 5K\lceil \log q \rceil + (2K + 2)\lceil \log N \rceil + 2K|h|$ bits. In the *winning price*

calculation process, each bidder needs to send $3K\lceil\log p\rceil$ bits for $\boldsymbol{\alpha'}_{P^i}, \boldsymbol{\beta'}_{P^i}$, $(\boldsymbol{\alpha'}_P(k))^{x_i}$, and $(2K+1)(2\lceil\log p\rceil + \lceil\log q\rceil)$ bits for the corresponding ZKPs, i.e., $(7K+2)\lceil\log p\rceil + (2K+1)\lceil\log q\rceil$ bits in total. Lastly, in the *winning bidder determination* process, each bidder sends $(n+2)\lceil\log p\rceil$ bits and $(n+2)(2\lceil\log p\rceil + \lceil\log q\rceil)$ bits, respectively, for distributed decryption and ZKPs, i.e., $(3n+6)\lceil\log p\rceil + (n+2)\lceil\log q\rceil$ bits in total. Therefore, total communication cost per bidder is $(2\lceil\log N\rceil + 15\lceil\log p\rceil + 7\lceil\log q\rceil + 2|h|)K + (3\lceil\log p\rceil + \lceil\log q\rceil)n + (6\lceil\log N\rceil + 10\lceil\log p\rceil + 4\lceil\log q\rceil)$ bits, and hence on the order of $O(n+K)$.

Therefore, the total communication complexity of each node in our scheme can be proved to be $O(n+K)$ bits, while that in [7] is $O(nK)$ bits.

3.6.2 Security and Privacy Analysis

We then investigate the security and privacy of the proposed auction framework SPA. We show as follows that SPA is secure and privacy-preserving not only under the honest-but-curious model, but also with regard to the malicious bidders who may want to deviate from the protocols to disrupt and/or learn more about the other bidders.

Theorem 8

[Privacy] A bidder's privacy is preserved regardless of the number of other colluding bidders. A seller's privacy is preserved too.

Proof: A bidder obtains a pair of public/private pseudo IDs in the identity initiation phase whenever he/she wants to participate in an auction. The bidder can use different such pseudo IDs for different auctions. Thus, the identity privacy can be preserved and the

bidder cannot be traced. Besides, our proposed auction scheme employs a distributed ElGamal cryptosystem. Unlike (n, k) threshold cryptosystems, where a ciphertext can be decrypted if k-out-of-n participants collude, our encryption scheme constructs a public key in an n-out-of-n secret sharing fashion. Therefore, a bidder's bidding vector encrypted with the public key can only be decrypted if the bidder participates in the distributed decryption. In our auction protocol, bidders do not collaboratively decrypt their own encrypted bidding vectors. They only jointly decrypt for $g^{\mathcal{P}(w)}$ as shown in (3.20), which is equal to 1 if p_w is the winning price and some random number otherwise. Thus, a bidder's bidding price privacy can also be preserved. Note that as mentioned before, we do not consider the possible adversaries at the DHT and Internet layers, since there have been several works addressing the privacy issues there [69, 70] and our DHT protocols can be easily adapted.

Similarly, since a seller can use different public pseudo IDs for different auctions, his/her identity privacy and trading history can be protected too. ■

Theorem 9

[Authenticity and Non-Repudiation] A bidder with legal pseudo IDs and legitimate bidding vectors can always be authenticated. Besides, a bid can be traced back to the bidder.

Proof: Due to the public pseudo ID authentication process, a malicious bidder cannot use a fake public pseudo ID or impersonate some other bidder to pass the authentication process according to Theorem 4 and 5. Legitimate bidding vectors can also be verified in the "ZKP generation" step of the bid encryption process. Besides, since each bidder signs his/her bids using an anonymous signature scheme based on their public and private pseudo IDs

as shown in the bid encryption process, a bid can be traced back to the bidder according to Theorem 6 and 7. ■

Theorem 10

[Auction Correctness] The proposed auction protocol results in correct outcomes with high probability (w.h.p.).

Proof: The winning price obtained from (3.20) will result in unintended outcomes if for some $1 \leq k \leq K$, we have $(\hat{B}_k - 3) \neq 0$ but $g^{(\hat{B}_k - 3) \cdot R_k} \equiv 1 \mod p$. Similarly, the winning bidder obtained from (3.22) will be incorrect if for some $1 \leq i \leq n$, we have $(\hat{b}_w^i - 2) \neq 0$ but $g^{(\hat{b}_w^i - 2) \cdot R_i} \equiv 1 \mod p$, where p_w is the winning price. However, since p is usually a very large number (e.g., 1024 bits), the probability of the occurrence of the above events is very low ($\approx 1/2^{1024}$). Therefore, the proposed auction protocol results in correct outcomes with high probability. ■

Theorem 11

[Auction Security] A malicious bidder deviating from the auction protocol cannot disrupt the auction outcome without being detected.

Proof: In our auction protocol, a malicious bidder may try to disrupt the auction outcome by indulging in the intermediate computations in the protocol. Notice that in the winning price calculation process, each bidder needs to submit ZKPs to prove that for each $1 \leq k \leq K$, $(\alpha_{P_k})^{R_k^i}$ and $(\beta_{P_k})^{R_k^i}$ are obtained using the same random value, and another ZKP to prove that $(\alpha'_{\mathbf{P}}(k))^{x_i}$'s ($1 \leq k \leq K$) are computed using the same x_i as in the key generation phase. In the winner bidder determination process, each bidder v_i submits a ZKP that $\alpha_{W^i}^{R_i}$, $\beta_{W^i}^{R_i}$ are computed using the same random number and another ZKP that

the x_j's used in computing $(\alpha_{W^i}^{R_i})^{x_j}$ $(1 \leq i \leq n)$ are the same as that used as in the distributed key generation process. Thus, any attempt to disrupt the auction outcome by deviating the protocol steps can be detected. ■

Table 3.3

Performance of our advertisement distribution scheme.

Soical Network Data Set	**Normalized Hop Count**
LiveJournal Social Network [71]	0.78
Astro Physics Collaboration Network [44]	0.75
Orkut Online Social Network [71]	0.71
Synthetic Data Set Using Nearest Neighbor (modified) Model [64]	0.64

3.7 Performance Evaluation

In this section, we evaluate the performance of the proposed secure and private auction framework SPA, particularly the advertisement distribution algorithm and the private auction scheme. In particular, we implement SPA on a PC with a Core i7 processor and 4GB RAM. We also implement the second price auction protocol [7]. In the experiments, the primes p and q in ElGamal cryptosystem are 1024 bits and 768 bits, respectively. The modulus N in the anonymous signature scheme is 1024 bits. Any hash function used in SPA results in digests of 128 bits.

We first evaluate the performance of our advertisement distribution algorithm. We implement it on several real social network graphs obtained from the SNAP project [3] as

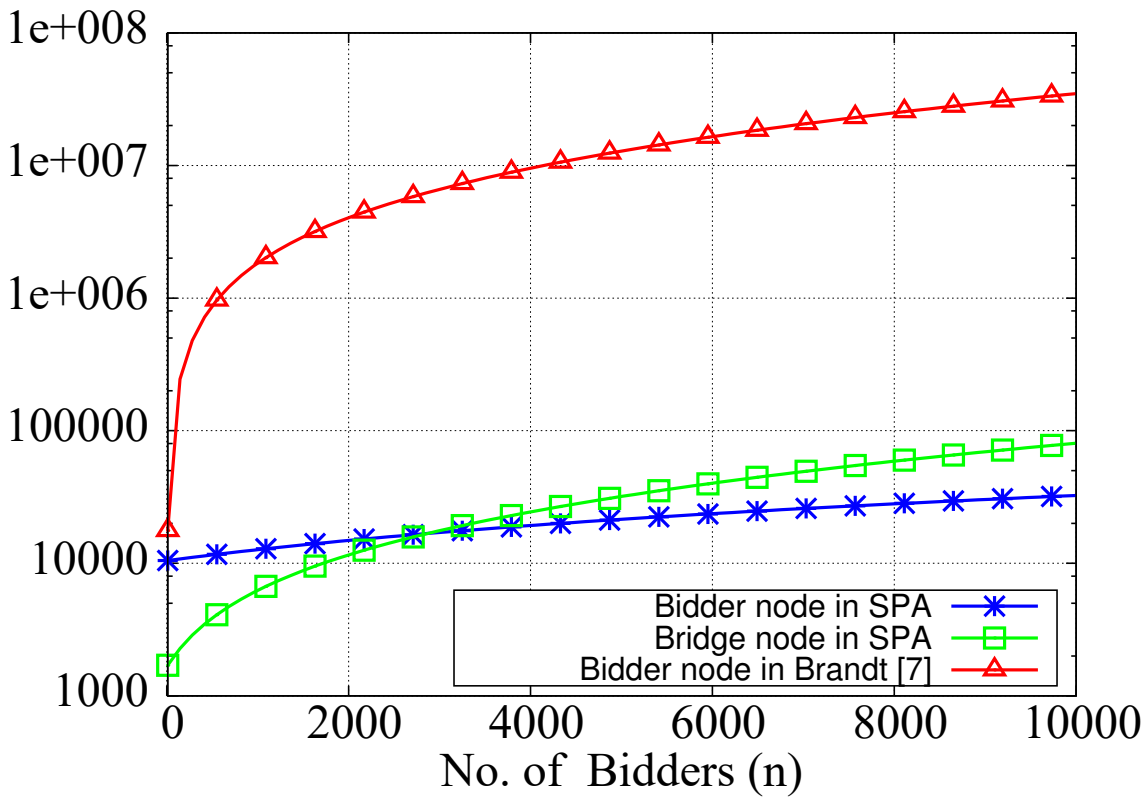

Figure 3.3

Comparison of Computation Cost (ms) when $K = 500$, $n \in [1, 10000]$

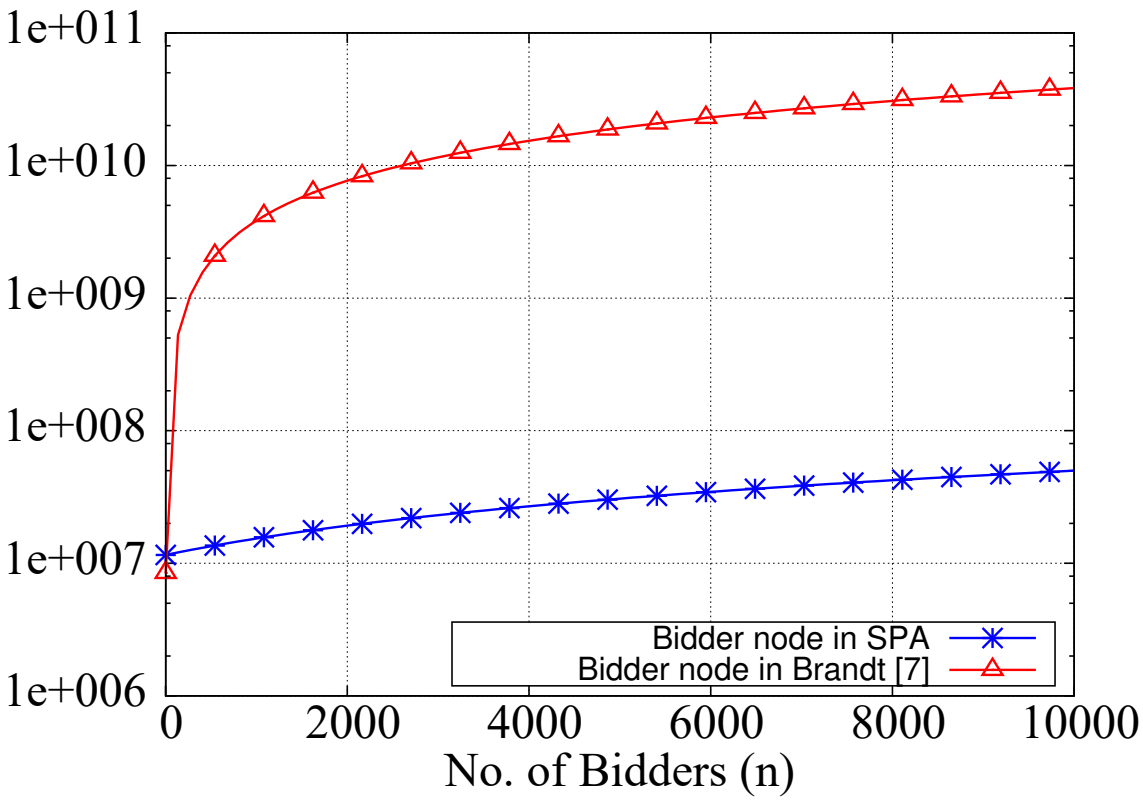

Figure 3.4

Comparison of Communication Cost (bits) when $K = 500$, $n \in [1, 10000]$

well as on a synthetic data set simulating OSNs. For each of these networks, we choose 100 random source and bridge node pairs and measure the average number of hops required to reach the bridge nodes. Table 3.3 shows the hop counts (normalized by $\log_2 n$ where n is the network size) based on different network data sets. The network data sets are listed in increasing order of average node degree. Intuitively, fewer hops are needed to deliver messages in networks with higher average node degrees. We can see that all advertisement messages can be delivered in $O(\log n)$ hops.

We then evaluate the performance of proposed the private auction scheme. We set the size of the price vector to 500. Lipmaa et al. [51] show that $k \leq 500$ suffices in most auctions in practice. Figure 3.3 compares the computation time of our protocol with that of [7]. We can see that the computation time of the bridge node and that of each bidder is well within the practical limits ($\sim$ 100 seconds) even when the number of bidders is large. In contrast, the computation time of a bidder in [7] is much higher (two orders of magnitude higher), and hence the protocol is impractical when the number of bidders is large. Likewise, as shown in Figure 3.4, the communication cost of a bidder in our protocol is much lower than that of a bidder in [7]. For example, in the case where there are 10000 bidders, the computation time of a bidder in our protocol is about 32.5 seconds and that of a bridge node is about 56.8 seconds, while the communication cost of a bidder is about 47MB. In the same case, the computation time of a bidder in [7] is more than 9.5 hours, and the communication cost of a bidder is about 35GB.

We also present the experiment results in Figure 3.5 and Figure 3.6 with different K's and n's. We can easily find that our auction protocol is much more efficient in terms of

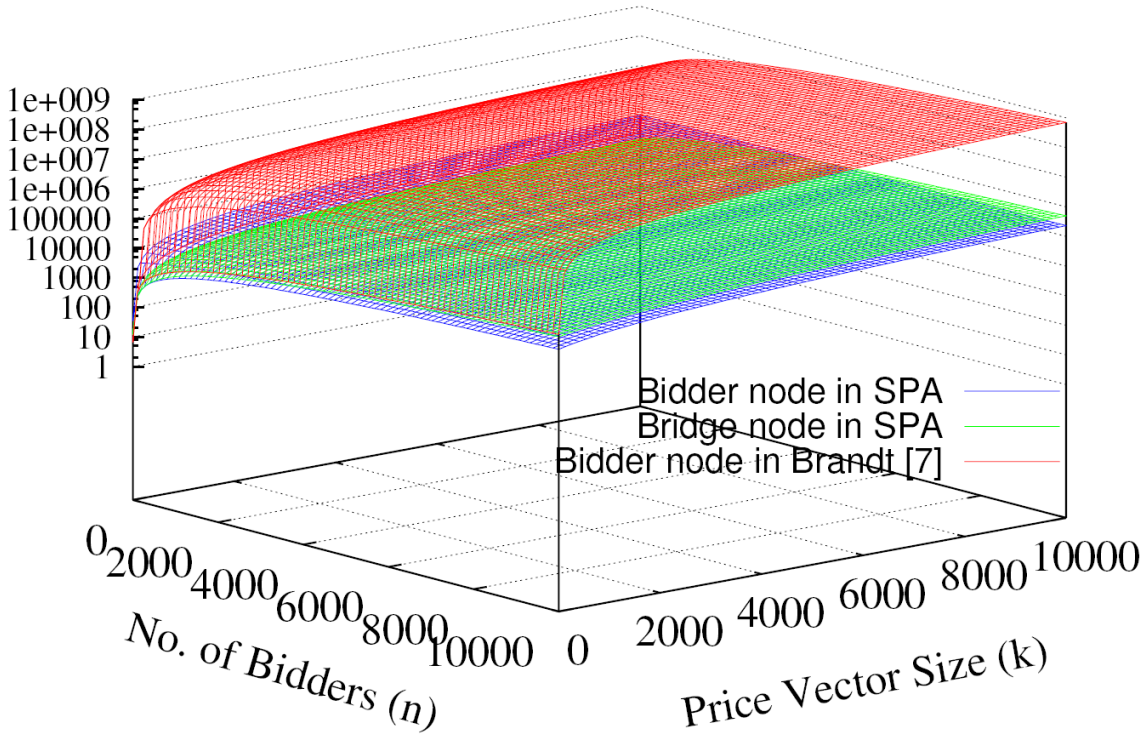

Figure 3.5

Comparision of Computation Cost (ms) when $n, K \in [1, 10000]$

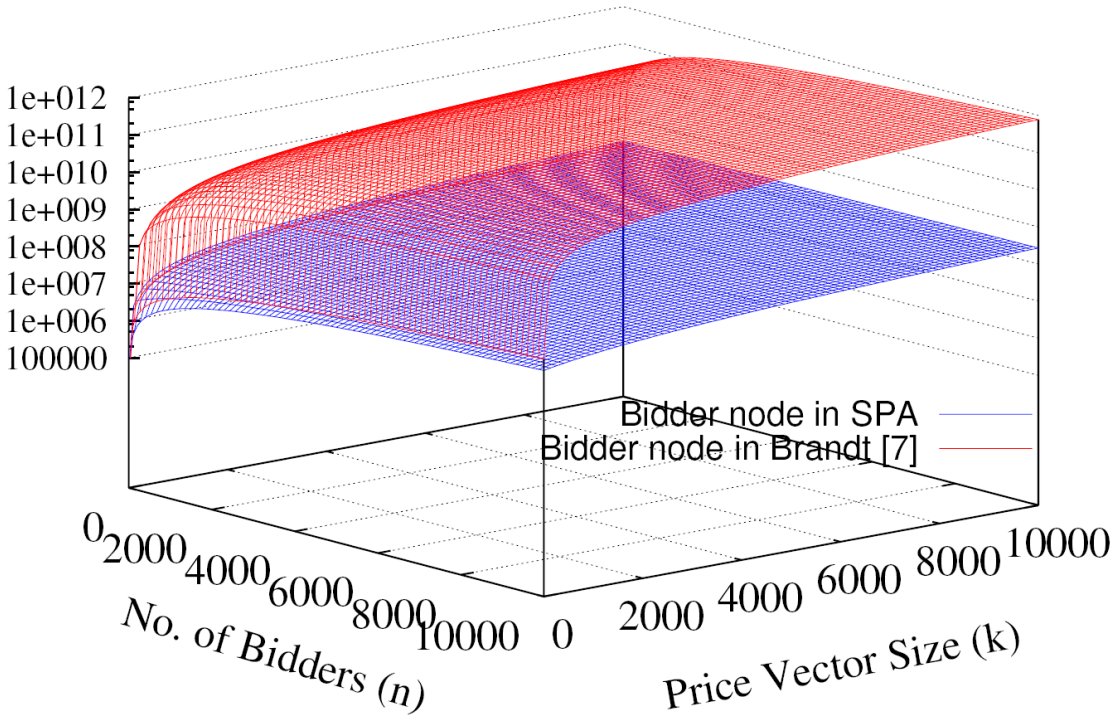

Figure 3.6

Comparison of Communication Cost (bits) when $n, K \in [1, 10000]$

both the computation cost and communication cost and is suitable for practical application even when both the size of the price vector and the number of bidders are large.

3.8 Conclusion

In this work, we have presented a secure and private auction framework, SPA, for DOSNs. System security can be protected against malicious attackers who try to disrupt auction outcomes by indulging in the intermediate computations in the protocol. Users' privacy, including their IDs and bidding prices, can also be guaranteed. In addition, SPA provides authenticity and non-repudiation, which are not made possible in previous auction schemes. The computation and communication complexities of our auction scheme are both $O(n+K)$. In contrast, the most efficient existing private auction schemes like [7] have complexities of $O(nK)$. Extensive experiment results have demonstrated the efficiency of the proposed framework.

CHAPTER 4

CONCLUSIONS AND FUTURE PLANS

In this dissertation, in Chapter 2, we proposed an asymmetric social proximity metric between two users in an OSN. The measure takes into account the user's own and his/her friends' weighted attributes (communities) while calculating social proximity. We then designed three cryptographic protocols to privately calculate social proximity between two users who may have different privacy requirements. In our privacy level 3 (L3P) protocol, each user is able to privately check if his/her proximity measure satisfies individually defined private threshold before establishing a friendship relationship and revealing more information about each other. The proposed asymmetric proximity model is validated using real social network data set. We also conduct extensive simulations to study the performance of the protocols and compare against existing state-of-the-art protocols.

In Chapter 3 we, for the first time, proposed a framework for secure and private auction based marketplace for decentralized online social network. The private auction protocol does not require the presence of third party (auctioneer) for the correctness and security of the auction. The participants in the auction scheme do not need to trust other participants for the auction, they trade anonymously, with authenticity and non-repudiation . We also proposed a distributed buyer seller matching algorithm to efficiently match buyer and sell-

ers for a given item and tested the efficiency of the algorithm using social network data sets. Our specific architecture of the framework makes it efficient enough to be applicable for the cases when there are large number of bidders. We verify the efficiency with extensive simulations.

As we increasingly generate and consume enormous amount of digital information, the security and privacy issues have become more important than ever. Future systems design needs to be secure not only against the usual malicious attacker(s), but also against the advanced persistent threats (APTs) and global surveillance capability of very resourceful entities. In the future, we will continue to work on these challenges and work on design and implementation robust, secure, and privacy friendly systems in the field of wireless network, complex networks, cyber physical system, and big data systems.

REFERENCES

[1] "Diaspora,", http://www.diasporaproject.org/.

[2] "QualNet Network Simulator,", http://web.scalable-networks.com/content/qualnet.

[3] "SNAP Project,", http://snap.stanford.edu/index.html, June 2013.

[4] "Alexa Top Sites,", http://www.alexa.com/topsites, May 2014.

[5] Z. Beerliová-Trubíniová and M. Hirt, "Perfectly-secure MPC with linear communication complexity," *Proceedings of the 5th conference on Theory of cryptography*, New York, USA, 2008.

[6] F. Brandt, "A Verifiable, Bidder-Resolved Auction Protocol," *Proceedings of the 5th International Workshop on Deception, Fraud and Trust in Agent Societies*, 2002.

[7] F. Brandt, "How to obtain full privacy in auctions," *Int. J. Inf. Secur.*, vol. 5, no. 4, 2006, pp. 201–216.

[8] S. Buchegger, D. Schiöberg, L.-H. Vu, and A. Datta, "PeerSoN: P2P social networking: early experiences and insights," *Proceedings of the SNS*, 2009.

[9] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy, "Blind and Anonymous Identity-Based Encryption and Authorised Private Searches on Public Key Encrypted Data," *Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography: PKC '09*, Irvine, CA, USA, March 2009.

[10] J. Camenisch and G. M. Zaverucha, " Private Intersection of Certified Sets," *Financial Cryptography and Data Security*, Accra Beach, Barbados, February 2009.

[11] D. Chaum and T. P. Pedersen, "Wallet Databases with Observers," *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, 1993.

[12] CNN, "Report: Eastern European gang hacked Apple, Facebook, Twitter,", http://www.cnn.com/2013/02/20/tech/web/hacked-apple-facebook-twitter/index.html, February, 2013.

[13] V. Conitzer, "Auction Protocols," *Algorithms and Theory of Computation Handbook*, 2nd edition, Chapman & Hall/CRC, 2009.

[14] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," *Proceedings of the EUROCRYPT*, 1997.

[15] E. D. Cristofaro, J. Kim, and G. Tsudik, "Linear-Complexity Private Set Intersection Protocols Secure in Malicious Model.," *16th International Conference on the Theory and Application of Cryptology and Information Security: ASIACRYPT'10*, Singapore, December 2010.

[16] E. D. Cristofaro, M. Manulis, and B. Poettering, "Private Discovery of Common Social Contacts," *Proceedings of the 9th international conference on Applied cryptography and network security: ACNS'11*, Nerja, Spain, June 2011.

[17] E. D. Cristofaro and G. Tsudik, "Practical Private Set Intersection Protocols with Linear Computational and Bandwidth Complexity,", Cryptology ePrint Archive, Report 2009/491, 2009.

[18] L. Cutillo, R. Molva, and T. Strufe, "Safebook: A privacy-preserving online social network leveraging on real-life trust," *IEEE Communications Magazine*, vol. 47, no. 12, 2009, pp. 94–101.

[19] I. Damgård, Y. Ishai, and M. Krøigaard, "Perfectly Secure Multiparty Computation and the Computational Overhead of Cryptography," *Proceedings of the 29th Annual international conference on Theory and Applications of Cryptographic Techniques: EUROCRYPT'10*, French Riviera, France, May 2010.

[20] I. Damgård, Y. Ishai, M. Krøigaard, J. B. Nielsen, and A. Smith, "Scalable Multiparty Computation with Nearly Optimal Work and Resilience," *Proceedings of the 28th Annual conference on Cryptology: Advances in Cryptology*, Santa Barbara, CA, USA, 2008.

[21] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-generation Onion Router," *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, Berkeley, CA, USA, 2004, SSYM'04, pp. 21–21, USENIX Association.

[22] W. Dong, V. Dave, L. Qiu, and Y. Zhang, "Secure Friend Discovery in Mobile Social Networks," *IEEE International Conference on Computer Communications (INFOCOM11)*, Shanghai, China, April 2011.

[23] J. Dreier, J.-G. Dumas, and P. Lafourcade, "Brandts Fully Private Auction Protocol Revisited," *Progress in Cryptology AFRICACRYPT 2013*.

[24] T. Elgamal, "A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, 1985, pp. 469–472.

[25] A. Fiat and A. Shamir, "How to prove yourself: practical solutions to identification and signature problems," *Proceedings on Advances in cryptology*, 1987, pp. 186–194.

[26] M. Franklin and M. Yung, "Communication Complexity of Secure Computation (extended abstract)," *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing: STOC'92*, Victoria, British Columbia, Canada, May 1992.

[27] M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold, "Keyword Search and Oblivious Pseudorandom Functions," *Proceedings of the Second international conference on Theory of Cryptography*, Cambridge, MA, USA, 2005.

[28] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient Private Matching and Set Intersection," *Proceedings of the 17th international conference on theory and application of cryptographic techniques: EUROCRYPT'04*, Interlaken, Switzerland, May 2004.

[29] L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40.

[30] R. Gennaro, M. O. Rabin, and T. Rabin, "Simplified VSS and Fast-track Multiparty Computations with Applications to Threshold Cryptography," *Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, Puerto Vallarta, Mexico, June 1998.

[31] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, New York, USA, 1987.

[32] L. C. Guillou and J.-J. Quisquater, "A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory," *EUROCRYPT*, 1988.

[33] M. Harkavy, J. D. Tygar, and H. Kikuchi, "Electronic auctions with private bids," *Proceedings of the WOEC*, Berkeley, CA, USA, 1998.

[34] C. Hazay and Y. Lindell, "Efficient Protocols for Set Intersection and Pattern Matching with Security against Malicious and Covert Adversaries," *Proceedings of the 5th conference on Theory of cryptography*, New York, USA, 2008.

[35] C. Hazay and K. Nissim, "Efficient Set Operations in the Presence of Malicious Adversaries," *Journal of cryptology*, vol. 25, no. 3, 2012, pp. 383–433.

[36] M. Hirt, U. M. Maurer, and B. Przydatek, "Efficient Secure Multi-party Computation," *Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security: ASIACRYPT'00*, Kyoto, Japan, December 2000.

[37] IGN, "Microsoft Hacked by Same Method as Apple and Facebook,", http://www.ign.com/articles/2013/02/23/microsoft-hacked-by-same-method-as-apple-and-facebook, February, 2013.

[38] S. Jarecki and X. Liu, "Efficient Oblivious Pseudorandom Function with Applications to Adaptive OT and Secure Computation of Set Intersection.," *Theory of Cryptography: TCC'09*, San Francisco, CA, USA, March 2009.

[39] A. Juels and M. Szydlo, "A Two-Server, Sealed-Bid Auction Protocol," *Financial Cryptography*, Lecture Notes in Computer Science, 2003, pp. 72–86.

[40] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, and D. Lewin, "Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web," *Proceedings of the STOC*, 1997, pp. 654–663.

[41] L. Katz, "A New Status Index Derived from Sociometric Analysis," *Psychometrika*, vol. 18, 1953.

[42] H. Kikuchi, "(M+1)st-Price Auction Protocol," *Proceedings of the Financial Cryptography*, 2002.

[43] L. Kissner and D. Song, "Private and Threshold Set-Intersection," *25th Annual International Cryptology Conference: CRYPTO'05*, Santa Barbara, California, USA, August 2005.

[44] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 1, 2007.

[45] C. Lesniewski-Laas and M. F. Kaashoek, "Whanau: a sybil-proof distributed hash table," *Proceedings of the NSDI*, 2010.

[46] M. Li, N. Cao, S. Yu, and W. Lou, "FindU: Privacy-preserving Personal Profile Matching in Mobile Social Networks," *IEEE International Conference on Computer Communications (INFOCOM11)*, Shanghai, China, April 2011.

[47] X. Liang, M. Barua, R. Lu, X. Lin, and X. Shen, "HealthShare: Achieving Secure and Privacy-preserving Health Information Sharing through Health Social Networks," *Computer Communications*, vol. 35, no. 15, 2012, pp. 1910–1920.

[48] D. Liben-Nowell and J. Kleinberg, "The Link Prediction Problem for Social Networks," *Proceedings of the twelfth international conference on Information and knowledge management*, New Orleans, LA, USA, 2003.

[49] Q. . W. M. Library, ,", http://rainbow.sunmoon.ac.kr/qualnet/ModelLibraries/QualNet-5.1-Wireless-ModelLibrary.pdf.

[50] H. Lin, S. S. M. Chow, D. Xing, Y. Fang, and Z. Cao, "Privacy-Preserving Friend Search over Online Social Networks,", Cryptology ePrint Archive, Report 2011/445, 2011.

[51] H. Lipmaa, N. Asokan, and V. Niemi, "Secure Vickrey auctions without threshold trust," *Proceedings of the Financial cryptography*, 2003.

[52] K. Liu, "Paillier's cryptosystem in Java,", http://www.csee.umbc.edu/ kunliu1/research/Paillier.html.

[53] R. Lu, X. Lin, X. Liang, and X. Shen, "A Secure Handshake Scheme with Symptoms-Matching for mHealthcare Social Network," *Mobile Networks and Applications*, vol. 16, 2011, pp. 683–694.

[54] P. Maymounkov and D. Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, 2002, IPTPS '01.

[55] G. L. Miller, "Riemann's Hypothesis and tests for primality," *Proceedings of the STOC*, 1975.

[56] M. Nagy, E. D. Cristofaro, A. Dmitrienko, N. Asokan, and A.-R. Sadeghi, "Do I know you?: efficient and privacy-preserving common friend-finder protocols and applications," *ACSAC*, 2013, pp. 159–168.

[57] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[58] M. Naor, B. Pinkas, and R. Sumner, "Privacy preserving auctions and mechanism design," *Proceedings of the 1st ACM conference on Electronic commerce*, 1999.

[59] D. H. Nyang and J. S. Song, "Knowledge-proof based versatile smart card verification protocol," *SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 3, July 2000.

[60] P. Paillier, "Public-key Cryptosystems based on Composite Degree Residuosity Classes," *Proceedings of the 17th international conference on theory and application of cryptographic techniques: EUROCRYPT'99*, Prague, Czech Republic, 1999.

[61] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," *Proceedings of the SIGCOMM*, 2001.

[62] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms*, 2001.

[63] K. Sako, "An Auction Protocol Which Hides Bids of Losers," *Proceedings of the PKC*, 2000.

[64] A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng, and B. Y. Zhao, "Measurement-calibrated graph models for social network experiments," *Proceedings of the World wide web*, 2010.

[65] C. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, 1991, pp. 161–174.

[66] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable peer-to-peer Lookup Service for Internet Applications," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, 2001, pp. 149–160.

[67] H. Tong, C. Faloutsos, and Y. Koren, "Fast Direction-aware Proximity for Graph Mining," *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, 2007.

[68] W. Vickrey, "Counterspeculation, Auctions and Competitive Sealed Tenders," *Journal of Finance*, 1961, pp. 8–37.

[69] Q. Wang and N. Borisov, "Octopus: A Secure and Anonymous DHT Lookup," *Proceedings of the IEEE ICDCS*, 2012.

[70] Q. Wang, P. Mittal, and N. Borisov, "In search of an anonymous and secure lookup: attacks on structured peer-to-peer anonymous communication systems," *Proceedings of the CCS*, 2010.

[71] J. Yang and J. Leskovec, "Defining and Evaluating Network Communities based on Ground-truth," *CoRR*, vol. abs/1205.6233, 2012.

[72] A. Yao, "How to Generate and Exchange Secrets," *27th Annual Symposium on Foundations of Computer Science*, October 1986.

[73] H. Yu, P. Gibbons, M. Kaminsky, and F. Xiao, "SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks," *IEEE Symposium on Security and Privacy*, 2008.

[74] L. Zhang, X.-Y. Li, and Y. Liu, "Message in a Sealed Bottle: Privacy Preserving Friending in Social Networks," *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, 2013, pp. 327–336.

[75] R. Zhang, Y. Zhang, J. S. Sun, and G. Yan, "Fine-grained Private Matching for Proximity-based Mobile Social Networking," *IEEE International Conference on Computer Communications (INFOCOM'12)*, Orlando, Florida, USA, March 2012.

[76] B. Zhao, H. Ling, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz, "Tapestry: a resilient global-scale overlay for service deployment," vol. 22, no. 1, 2004, pp. 41–53.

[77] H. Zhu, S. Du, M. Li, and Z. Gao, "Fairness-Aware and Privacy-Preserving Friend Matching Protocol in Mobile Social Networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, June 2005, pp. 192–200.